AFWAL-TR-85-1016 VOL III

EVALUATION & VALIDATION (E&V)

TEAM PUBLIC REPORT, Volume III

RAYMOND SZYMANSKI
E&V TEAM CHAIRMAN
AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6543

SEPTEMBER 1987

Interim Technical Report for Period
1 October 1985 - 30 June 1987

APPROVED FOR PUBLIC RELEASE,
DISTRIBUTION UNLIMITED

PREPARED FOR:

ADA* JOINT PROGRAM OFFICE
3D139 (FERN ST/C107) PENTAGON
WASHINGTON, D.C. 20301

AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6543

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for Public Release;
Distribution Unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFWAL-TR-85-1016, Vol III

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Air Force Wright
Aeronautical Laboratories

6b. OFFICE SYMBOL *(If applicable)*
AFWAL/AAAF-2

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Wright-Patterson Air Force Base, OH 45433-6543

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Ada*JOINT PROGRAM OFFICE

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*
3D139 (Fern Street/C107) Pentagon
Washington, D. C. 20301

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.
63226	AJPO	28	53

11. TITLE *(Include Security Classification)*
Evaluation and Validation (E&V) Team Public Report, Volume III (U)

12. PERSONAL AUTHOR(S)
Raymond Szymanski, E&V Team Chairman

13a. TYPE OF REPORT
Interim Technical

13b. TIME COVERED
FROM 1 Oct 85 TO 30 Jun 87

14. DATE OF REPORT *(Year, Month, Day)*
1987 September

15. PAGE COUNT
279

16. SUPPLEMENTARY NOTATION
*Ada is a Registered Trademark of the U.S. Government (Ada Joint Program Office)

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
09	02	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Ada*
Evaluation
Validation
Common APSE Interface Set (CAIS)
Ada Programming Support Environment

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Activities and accomplishments of the Evaluation and Validation (E&V) Team are reported for FY1986-87. The purpose of the E&V Task, which is sponsored by the Ada Joint Program Office, ~~(AJPO)~~, is to develop techniques and tools that will provide a capability to perform assessment of Ada Programming Support Environments (APSEs) and to determine conformance of APSEs to the Common APSE Interface Set (CAIS). As this technology is developed, it is being made available to DoD components, industry, and academia.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Raymond Szymanski

22b. TELEPHONE *(Include Area Code)*
(513) 255-2446

22c. OFFICE SYMBOL
AFWAL/AAAF-2

DD Form 1473, JUN 86 *Previous editions are obsolete.*

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs, (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Raymond Szymanski | 20 April 1988
RAYMOND SZYMANSKI | Date
Project Engineer

FOR THE COMMANDER

Charles H Krueger Jr | 21 Apr 88
CHARLES H. KRUEGER, JR | Date
Acting Chief
System Avionics Division

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed at your organization please notify AFWAL/AAAF, WPAFB, OH 45433-6543 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

TABLE OF CONTENTS

SECTION I - Project Technical Summary 1-1

APPENDIX A - Evaluation and Validation Plan, Version 4.0 . . . . . . A-1

APPENDIX B - Issues and Strategies for CAIS Evaluation and Validation, Version 1.0 . . . . B-1

APPENDIX C - Tools and Aids Document, Version 1.0 . . . . C 1

APPENDIX D - Requirements for the Evaluation and Validation of Ada Programming Support Environments, Version 2.0 . . D-1

APPENDIX E - Minutes of the Evaluation and Validation (E&V) Meeting December 1985 . . . . E-1

APPENDIX F - Minutes of the Evaluation and Validation (E&V) Meeting March 1986 . . . . F-1

APPENDIX G - Minutes of the Evaluation and Validation (E&V) Meeting June 1986 . . . . G-1

APPENDIX H - Minutes of the Evaluation and Validation (E&V) Meeting September 1986 . . . . H-1

APPENDIX I - Minutes of the Evaluation and Validation (E&V) Meeting December 1986 . . . . I-1

APPENDIX J - Minutes of the Evaluation and Validation (E&V) Meeting March 1987 . . . . J-1

APPENDIX K - Minutes of the Evaluation and Validation (E&V) Meeting June 1987 . . . . K-1

LIST OF FIGURES

Figure A-1 E&V MANAGEMENT STRUCTURE A-14

Figure A-2 E&V RELATIONSHIP TO OTHER ORGANIZATIONS A-20

Figure A-3 APSE E&V TASK WORK BREAKDOWN STRUCTURE A-26

Figure A-4 MAPPING OF WBS ELEMENTS TO E&V OBJECTIVES A-27

SECTION I

PROJECT TECHNICAL SUMMARY

1.0 INTRODUCTION

This report is the third in a series of technical reports to be published by the Evaluation and Validation (E&V) Team. The purpose of the E&V Public Report is to provide an overview of the many technical accomplishments of the E&V Team during an appropriate time frame. This third report contains information resulting from E&V activities during October 1985 to September 1987 which is being made available for public review and comment. Contents of this report reflect an observation of the E&V Team progress during the specified time frame and should not be viewed as final representations of the technology being developed.

1.2 Background

In June 1983 the Ada Joint Program Office proposed the formation of the E&V Task and a tri-service APSE E&V Team, with the Air Force designated as lead service. In October 1983 the Air Force officially accepted responsibility as lead service on the E&V Task .

The Ada community, including government, industry, and academic personnel, needs the capability to assess APSEs (Ada Programming Support Environments) and their components and to determine their conformance to applicable standards (e.g., DOD-STD-1838, the CAIS standard). The technology required to fully satisfy this need is extensive and largely unavailable; it cannot be acquired by a single government sponsored, professional society sponsored, or private effort. The purpose of the Evaluation and Validation (E&V) Task is to provide a focal point for addressing the need by:

(1) identifying and defining specific technology requirements,

(2) developing selected elements of the required technology,

(3) encouraging others to develop some elements, and

(4) collecting information describing existing elements.

This information will be made available to DoD components, other government agencies,

1.3 E&V Meetings

E&V Team meetings are held on a quarterly basis. The following meetings were held at Wright-Patterson Air Force Base in Dayton, Ohio: 5-7 March 1986, 2-4 June 1986, 3-5 September 1986, 4-6 March 1987, 3-5 June 1987. The 1-3 December 1986 Meeting was held in San Diego, California.

1.4 E&V Team Organization

In order to coordinate all of the activities to be accomplished within the E&V Task, the E&V Team is partitioned into six working groups. The identification of these working groups, and their associated areas of responsibility, are delineated in the following sections. These working groups are subject to change during the life of the E&V Task. Each working group has a designated Chairperson and Vice-Chairperson. It is the responsibility of each working group Chairperson to coordinate the activities of the working group with the E&V Team Chairperson. In addition, each working group Chairperson is required to brief the status of the respective working group at every E&V Team meeting.

1.4.1 Directional Management Working Groups

1.4.1.1 E&V Requirements Working Group (REQWG)

The REQWG is responsible for performing the following tasks:

- Maintain an E&V Requirements Document against which the E&V Reference Manual will be developed.

- Provide analysis of requirements in the E&V Requirements Document to determine their adequacy, completeness, traceability, testability, consistency, and feasibility.

- Identify issues which may impact the development of E&V technology.

- Provide recommendations for acquisition of E&V tools and aids through the development of an E&V Tools and Aids Document.

- Prepare position papers through the duration of the E&V Task which address issues on E&V requirements.

1.4.1.2 E&V Standards Evaluation and Validation Working Group (SEVWG)

The SEVWG is responsible for performing the following tasks:

- Recommend specific areas of consideration for standards related to future evaluations and validations.

- Emphasize study on the CAIS.

- Review the development of the CAIS and identify areas of possible concern to E&V.

- Provide presentations to the E&V Team on the CAIS.

- Provide liaison activity to the KIT.

- Prepare position papers throughout the duration of the E&V Task which address particular aspects of the CAIS as relevant to E&V.

1.4.1.3 E&V Coordination Working Group (COORDWG)

The COORDWG is responsible for performing the following tasks:

- Develop a Technical Coordination Strategy Document which will:

    * identify related technical efforts;

    * identify relationships between the E&V Task and each of the related tasks;

    * identify areas of mutual benefit to the tasks;

    * identify impact of schedules;

    * identify level of coordination required;

    * identify issues which require resolution to the mutual benefit of the tasks involved.

- Identify professional organizations which are technically related to the E&V effort.

- Develop a Public Coordination Strategy Document which provides an approach as to how such public coordination will be performed.

- Maintain and distribute a set of E&V viewgraphs and corresponding text to allow E&V Team members to present the status of the E&V Task at public meetings.

- Prepare E&V status reports for publication in related journals and newsletters and dissemination at related conferences.

- Catalog all issues related to the E&V effort.

- Develop and maintain an E&V Project Reference List.

1.4.2 Technical Management Working Groups

1.4.2.1 E&V Ada Compiler Evaluation Capability Working Group (ACECWG)

The ACECWG is responsible for performing the following tasks:

- Provide a formal interface between the Ada community and the ACEC effort.

- Evaluate and critique aspects of the technical approach being employed on the ACEC effort.

- Evaluate and critique selected ACEC deliverables.

- Discuss and provide feedback on issues critical to the ACEC.

1.4.2.2 E&V CAIS Validation Capability Working Group (CVCWG)

The E&V CVCWG is responsible for performing the following tasks:

- Provide technical expertise to E&V chairman and team for review of CVC contractors' products and activities.

- Provide to E&V chairman and CVC project engineer recommendations regarding validation of CAIS.

- Coordinate regularly and closely with SEVWG concerning validation of DoD Standard 1838 implementations.

1.4.2.3 E&V Technology Classification Working Group (CLASSWG)

The CLASSWG is responsible for performing the following tasks:

- Serve as focal point for analysis of Reference System (Reference Manual and Guidebook).

- Solicit information and recommendations regarding E&V technology.

- Classify E&V technology.

- Aid in the technology transition of the Reference System.

- Delineate whole APSE issues.

- Recommend new areas of investigation.

1.5 Conclusion

This E&V Public Report is being made available by the E&V Team in order to solicit comments from those individuals who are not actively involved in the E&V Task. All comments should be addressed to:

Raymond Szymanski
AFWAL/AAAF
Wright-Patterson AFB, Ohio 45433-6543
(SZYMANSK@AJPO.SEI.CMU.EDU or
EV-TEAM@AJPO.SEI.CMU.EDU)

APPENDIX A

EVALUATION AND VALIDATION
(E&V)

PLAN

VERSION 4.0

4 JUNE 1987

The Task for the Evaluation & Validation of Ada* Programming Support Environments (APSEs) is sponsored by the Ada Joint Program Office(AJPO).

*Ada is a Registered Trademark of the U.S. Government (Ada Joint Program Office)

Table of Contents

1.0 INTRODUCTION A-4
1.1 Objective of the E&V Plan A-4
1.2 Background A-5

2.0 SCOPE A-6

3.0 E&V TECHNICAL APPROACH A-9
3.1 APSE Concept A-9
3.2 APSE E&V Classification Schema A-10
3.2.1 Step 1: Identification of APSE Components A-10
3.2.2 Step 2: Identification of APSE Attributes A-10
3.2.3 Step 3: Identification of APSE E&V Categories . . A-11

4.0 E&V MANAGEMENT APPROACH A-12
4.1 Ada Joint Program Office A-12
4.2 Air Force, Army, Navy A-12
4.3 E&V Team Chairperson A-12
4.4 E&V Team A-15
4.5 E&V Team Working Groups A-15
4.5.1 Directional Management Working Groups A-16
4.5.1.1 E&V Requirements Working Group (REQWG) . . A-16
4.5.1.2 E&V Standards Evaluation and Validation Working Group (SEVWG) A-16
4.5.1.3 E&V Coordination Working Group (COORDWG) . A-16
4.5.2 Technical Management Working Groups A-17
4.5.2.1 E&V Ada Compiler Evaluation Capability Working Group (ACECWG) A-17
4.5.2.2 E&V CAIS Validation Capability Working Group (CVCWG) A-17
4.5.2.3 E&V Technology Classification Working Group (CLASSWG) A-18
4.6 Contractor Support A-18
4.6.1 Ada Compiler Evaluation Capability A-18
4.6.2 CAIS Validation and Evaluation A-18
4.6.3 Technical Support A-18

5.0 E&V RELATIONSHIP TO OTHER ORGANIZATIONS A-18
5.1 KIT A-19
5.2 User Groups and Professional Societies A-19
5.3 Standards Organizations A-19
5.4 Ada Board A-19

6.0 E&V DELIVERABLES A-21

7.0 E&V WORK BREAKDOWN STRUCTURE A-25
7.1 1000 APSE E&V Management A-28
7.2 2000 APSE E&V Requirements A-28
7.3 3000 APSE E&V Reference Manual Development A-29
7.4 4000 APSE Evaluation Capability A-30
7.5 5000 APSE Validation Capability A-31

7.6 6000 APSE E&V Tools/Aids A-31
7.7 7000 APSE E&V Support A-33

8.0 E&V SCHEDULES/MILESTONES A-33
8.1 E&V Milestones Accomplished A-33
8.2 E&V Milestones Scheduled A-34

9.0 E&V REFERENCES A-35

APPENDIX A Acronyms A-36

1.0 INTRODUCTION

1.1 Objective of the E&V Plan

The purpose of the E&V Plan is to provide a detailed and organized approach to the development of technology which will be used as a basis for the Evaluation and Validation (E&V) of Ada Programming Support Environments (APSEs). The E&V Plan will be updated on an annual basis throughout the duration of the E&V Task. Version 3.0 of the E&V Plan, dated 13 June 1986, was used to provide technical guidance to the E&V Teams during the third year of the E&V Task. This current version of the E&V Plan contains modifications to Version 3.0 which reflect some changes in the E&V Team, revisions to the schedule, and provides a list of accomplishments since the last E&V Plan.

This document is organized as follows:

- Section 1: INTRODUCTION

  * Section 1 presents: (1) the objective of the E&V Plan; and (2) historical background information which led to the establishment of the E&V Team.

- Section 2: SCOPE

  * Section 2 presents the scope of the E&V Task through delineation of the E&V Task objectives.

- Section 3: E&V TECHNICAL APPROACH

  * Section 3 provides an overview of the technical approach to the development of E&V technology by defining an E&V Classification Schema.

- Section 4: E&V MANAGEMENT APPROACH

  * Section 4 provides the management structure for the E&V Task and identifies specific tasks for Working Groups within the E&V Team.

- Section 5: E&V RELATIONSHIP TO OTHER ORGANIZATIONS

  * Section 5 describes the relationship of the E&V Task to other DoD and technical organizations.

- Section 6: E&V DELIVERABLES

  * Section 6 presents a description of all of the deliverables expected from the E&V Task.

- Section 7: E&V WORK BREAKDOWN STRUCTURE

  * Section 7 presents a Work Breakdown Structure which delineates all of the activities to be accomplished in the E&V Task.

- Section 8: E&V SCHEDULES/MILESTONES

  * Section 8 presents schedules and milestones associated with the E&V Task.

- Section 9: E&V REFERENCES

  * Section 9 provides a list of references which are used within this document.

1.2 Background

In 1975 the High Order Language Working Group (HOLWG) was formed under the auspices of the U.S. DoD. It consisted of representatives from the Army, Air Force, Navy, Marines and other defense agencies, with the goal of establishing a single high order language for new DoD Embedded Computer Systems (ECS). The technical requirements for the common language were finalized in the STEELMAN [1] report of June 1978. International competition was used to select the new common language design. In 1979, after review by approximately eighty teams (representing DoD organizations, industry, academia and NATO countries), and after intensive analysis by the three Services and other defense agencies, the DoD selected the design developed by Jean Ichbiah and his colleagues at CII-Honeywell Bull. The language was named Ada in honor of Agusta Ada Byron (1815-1851), the daughter of Lord Byron and the first computer programmer.

Early in the development process it was realized that the acceptance and the benefits derived from a common language could be increased substantially by the development of an integrated system of software development and maintenance tools. The requirements for such an Ada programming environment were stated in the STONEMAN [2] document. STONEMAN identifies the APSE as support for "the development and maintenance of Ada application software throughout its life cycle."

The Army has completed development of an APSE known as the Ada Language System (ALS). The Navy is in the process of procuring an APSE development which will be based upon the Army's APSE and will be known as the Ada Language System/Navy (ALS/N).

The Ada Joint Program Office (AJPO) was formed in December 1980. It is the principal DoD agent for development, support and distribution of tools, common libraries, and coordination of Ada. The AJPO will coordinate all Ada efforts within DoD to ensure their compatibility with the requirements of other Services and DoD agencies, to avoid duplicative efforts and to maximize sharing of resources.

The KAPSE Interface Team (KIT), a tri-service organization which is chaired by the Navy under the guidance of the AJPO, was established in late 1981 as the result of a Memorandum of Agreement (MOA) signed by the Deputy Under Secretary of Defense and the Assistant Secretaries of the three services. The objective of the KIT is to define a standard set of Kernel Ada Programming Support Environment (KAPSE) interfaces to ensure the interoperability of data and the transportability of tools between conforming APSEs. The Common APSE Interface Set (CAIS) developed by the KIT provides the virtual operating system on which tools run, as well as the minimum set of command, edit and similar functions required to transport tools from one CAIS to another. The KAPSE Interface Team from Industry and Academia (KITIA) was established in early 1982. The KITIA consisted of volunteer representatives from industry and universities who provided expertise relevant to the definition of the CAIS.

In addition to the KIT/KITIA development of the CAIS, which will require the development of a validation capability to determine conformance, other efforts have contributed to the foundation of the E&V Task. One such effort was the formation of the Ada Validation Organization (AVO), under the direction of the AJPO. The AVO is responsible for the development of an Ada Compiler Validation Capability (ACVC) which is currently used to ensure that Ada compiler developers have correctly implemented the standard Ada language (ANSI/MIL-STD-1815A-1983). A second effort which is fundamental to the E&V task is the National Bureau of Standards' Taxonomy for an APSE [3], which systematically defines tool capabilities for a full APSE. A third effort, at the Air Force Wright Aeronautical Laboratories [4], provided an initial APSE evaluation questionnaire that can be used as a baseline from which to develop a more refined, comprehensive, and generic set of questions. Finally, previous and current efforts, sponsored by the AJPO, at Virginia Polytechnic Institute and State University [5], Arizona State University, and the Institute for Defense Analysis have addressed issues associated with validation in APSEs.

In June 1983 the AJPO proposed the formation of the E&V Task and a tri-service APSE E&V Team, with the Air Force designated as lead service [6]. In October 1983 the Air Force officially accepted responsibility as lead service on the E&V Task [7].

2.0 SCOPE

The Ada community, including government, industry, and academic personnel, needs the capability to assess APSEs (Ada Programming Support Environments) and their components and to determine their conformance to applicable standards (e.g., DOD-STD-1838, the CAIS standard). The technology required to fully satisfy this need is extensive and largely unavailable; it cannot be acquired by a single government sponsored, professional society sponsored, or private effort. The purpose of the Evaluation and Validation (E&V) Task is to provide a focal point for addressing the need by (1) identifying and defining specific technology requirements, (2) developing selected elements of the required

technology, (3) encouraging others to develop some elements, and (4) collecting information describing existing elements. This information will be made available to DoD components, other government agencies, industry, and academia.

In order to accomplish the purpose of the E&V Task, nine specific objectives have been identified. Note that each objective is preceded by "O-" (indicating Objective) and a unique number. This nomenclature is provided to enable illustration of a direct mapping of the E&V Work Breakdown Structure elements (provided in Section 7) to the following specific objectives:

- O-1: Develop Requirements for APSE E&V

  * As a prerequisite to the development of APSE E&V technology, E&V requirements must be specified. The development of E&V requirements will be based upon examination of APSE related issues such as life-cycle methodologies, human engineering aspects, software engineering practices, etc. The E&V requirements which are developed will be used to guide the E&V technical effort.

- O-2: Develop APSE E&V Classification Schema

  * The technical approach to classifying APSE components will be based upon an APSE E&V Classification Schema. This schema is comprised of three major factors: (1) identification of APSE components; (2) identification of associated APSE attributes for each APSE component; and (3) identification of the appropriate evaluation or validation capability associated with each APSE component. Section 3 (E&V TECHNICAL APPROACH) of this document provides additional detail on the APSE E&V Classification Schema which will be used by the E&V Task. This schema will be refined during the E&V Task.

- O-3: Identify and Classify APSE Components

  * APSE components will be identified and classified based upon the existence of criteria and standards as well as the existence of metrics capabilities for those components. The identification and classification of APSE components will be in accordance with the APSE E&V Classification Schema.

- O-4: Develop APSE Evaluation Capability

  * An evaluation capability will be developed for all APSE components for which there exist no formal standards (i.e., MIL-STD, ANSI, etc.). The evaluation capability for some components will be provided through established metrics, whereas the evaluation capability for other components may be limited to a detailed questionnaire.

* As a first step toward achieving this objective, previous AFWAL efforts in the area of APSE evaluation will be reviewed for applicability as a baseline for generic evaluation criteria. Because evaluation criteria will be largely dependent upon the defined functionality of each tool, an analysis will be made of the functionality of various tools provided in the DoD APSEs to determine commonality among tool names and tool functions. This analysis will be closely coordinated with the National Bureau of Standards (NBS) effort in defining a taxonomy of APSE tool features. Ongoing standards development activities will be reviewed as a potential source of evaluation criteria and public presentation of the findings of the analysis will be used to solicit input from industry and academia so as to generate a sound and realistic expansion of the developed criteria.

- O-5: Develop APSE Validation Capability

  * A validation capability will be developed for the proposed MIL-STD 1838, which has been developed by the KIT/KITIA. If other APSE related standards are established (i.e., 1838A) appropriate validation capabilities will be developed. Examination of the current validation procedures and Ada Compiler Validation Capability (ACVC) test suite utilized by the Ada Validation Organization (AVO), as well as procedures implemented by ANSI and ISO, will be used as a baseline. The CAIS operational definition work at Arizona State University will provide an available baseline from which a validation capability may be developed.

- O-6: Develop Evaluation & Validation Tools and Aids

  * As the requirements for E&V are determined, various software tools/aids will be identified as essential to the E&V effort. Such tools/aids include test sets, test scenarios, data reduction capability, and other designated means of automated support. As these tools/aids become more clearly defined, an assessment will be made to include such capability. Existing tools/aids which are applicable to the E&V Task will be considered for use. New tools/aids which are determined to be essential for the APSE E&V Task will be assessed for possible contractor development. One specific validation capability which will be developed through a contractual effort will be the CAIS Validation Capability (CVC). The existing Ada Compiler Validation Capability (ACVC) will be included as part of the E&V Tools/Aids.

- O-7: Develop Procedures for Implementation of E&V

  * The E&V Task will develop and provide the technology and procedures by which E&V of APSEs will be accomplished; however, it will not provide an E&V Organization which will be

responsible for the execution of evaluation and validation procedures on all APSEs. The E&V procedures will be based upon E&V requirements, APSE standards, evaluation criteria, validation capability, and existing E&V tools/aids. Once the procedures and mechanisms are fully developed, the APSE Validation execution responsibility will be transitioned to an appropriate validation organization. The APSE Evaluation capability will be transitioned to the community for use by DoD components, industry, and academia.

- O-8: Provide Initiative and Focal Point With Respect to APSE E&V

  * There currently exists a need to provide a focal point for APSE developers and users with regard to information about E&V of APSEs. APSE E&V questions arise frequently within professional societies and user groups. A forum is needed in which APSE E&V questions can be addressed and discussed, and in which APSE E&V information can be disseminated throughout the Ada community.

  * The E&V Team will provide a focal point for APSE E&V for the Ada community. Public reports on the results of this APSE E&V Plan will be made available to professional organizations such as SIGAda and AdaJUG. This is in keeping with the AJPO philosophy of public dissemination of information. The E&V task is the lead DoD effort with regard to APSE E&V. In this respect, the E&V Team will participate in, and assist where possible, other programs technically related with APSE E&V. Such programs include the KIT/KITIA, the Ada Validation Organization, and international development efforts.

- O-9: Promote Community Use and Acceptance of the E&V Effort

  * Use of the E&V technology developed through this task will provide for an orderly progression of technology insertion into environments. The E&V technology thus developed will be extendible to other software development efforts, thereby maximizing the economic benefits of the E&V task products and minimizing the cost within DoD and industry of doing E&V related work.

3.0 E&V TECHNICAL APPROACH

3.1 APSE Concept

The APSE, as depicted by the STONEMAN document, provides a virtual interface between the user of the APSE and the particular host system upon which the APSE is installed. This interface is designed to be machine and operating system independent; in effect, it defines an Ada virtual machine whose features are available on all actual host machines. The purpose of the APSE is to provide an environment for the design, development, documentation, testing, management, and

maintenance of embedded computer software, written principally in the Ada Programming language.

The initial efforts of the E&V Task were based upon the concept of an APSE structure as defined by the original STONEMAN document. STONEMAN paints a broad picture of the needs and identifies the relationships of the parts of an integrated APSE.

3.2 APSE E&V Classification Schema

The technical approach to the E&V effort requires that APSE components be identified and classified based upon a well-defined Classification Schema. The schema creates a framework for a logical sequence of steps leading to the definition of elements of E&V technology, as follows:

- Step 1: Identification of APSE Components;

- Step 2: Identification of APSE Attributes; and

- Step 3: Identification of APSE E&V Categories.

The following sections present additional detail on each of these steps and is expected to influence the organization of the E&V Reference Manual. The E&V Classification Schema which is presented in this document is expected to be further refined during the E&V Task.

3.2.1 Step 1: Identification of APSE Components

For the purpose of the E&V Classification Schema, APSE components are defined to be features of the APSE. The National Bureau of Standards Taxonomy of Tool Features for the APSE [3] presents a hierarchical arrangement of software tool features. The first (highest) level is an abstract level which encompasses all of the features below it. The second level includes the basic processes of the APSE (i.e., input, output, and function). The third level includes the classes of tool features (i.e., subject, control, transformation, static analysis, dynamic analysis, management, user output, and machine output). The fourth and fifth levels include specific APSE features.

Initially, as a basis for Step 1, the functional part of current APSE functional taxonomies will be used to identify APSE components. However, as additional E&V Requirements are specified during the E&V Task, the list of APSE functions will be expanded to reflect: (1) additional APSE features; and (2) finer granularity of previously identified APSE features.

This first step of the Classification Schema results in a hierarchical structure which can be illustrated by a list of APSE functions, identified through an appropriate numbering scheme.

3.2.2 Step 2: Identification of APSE Attributes

Following the development of a functional taxonomy for the classification of APSE components, an attribute taxonomy will be developed. Meaningful function/attribute pairs will be identified as key aspects of E&V component assessment objectives. Other, functionally independent attributes will be identified as aspects of component or entire APSE assessment objectives.

3.2.3 Step 3: Identification of APSE E&V Categories

For the purpose of the E&V Classification Schema, the term "Evaluation" represents a method of assessing the quality of APSE components for which no specific standard (i.e., MIL-STD, ANSI, etc.) exists, or for which a standard may exist but there is no known capability to measure conformance to that standard. The term "Validation" represents a method of determining conformance to a standard which is applicable to an APSE (e.g., MIL-STD 1815A, CAIS, etc.).

The determination of what methodology (i.e., evaluation or validation) is then based on whether a standard exists and whether a means of checking conformance to that standard also exists. Different levels of conformance checking exist and that leads to a partitioning of validation methodology into non-formal and formal techniques. Based on this notion of standards and conformance checking, the following categories are provided for determining appropriate assessment methodology.

- Category A:

  * If no standard for an APSE component exists and no technique of evaluating conformance has been developed, then the component requires subjective evaluation.

- Category B:

  * If no standard for an APSE component exists, but a method for assessing the quality (i.e., a metrics capability) exists, then the component requires objective evaluation.

- Category C:

  * If a standard for an APSE component exists but there is no existing method for determining conformance to that standard then the component is in an intermediate category.

- Category D:

  * If both a standard for an APSE component and a method for determining conformance to that standard exist, then the component requires validation.

- Category E:

  * If a standard for an APSE component and a purely formal technique for determining conformance to that standard exist, then the component requires formal validation.

When these categories are applied to APSE components the appropriate quality assessment technology for each component type may be easily determined.

As the third step in the E&V Classification Schema, each APSE function/APSE attribute couple will be examined to determine which APSE E&V Category is most appropriate, based upon existing standards/criteria and metrics capabilities.

The result of associating APSE functions with relevant APSE attributes and E&V categories is primarily to determine what standards and assessment techniques have to be developed in an independent manner. In other words, the E&V Classification Schema allows the decision to pursue the development of standards, validation methods, or formal methods independently of what course may be chosen for other components even in the same area of application.

4.0 E&V MANAGEMENT APPROACH

Figure A-1, page A-14, depicts the E&V management structure. Each of the components is identified in the following sections.

4.1 Ada Joint Program Office

The Ada Joint Program Office (AJPO) sponsors the E&V Task. All E&V Team activities are coordinated with the AJPO through the E&V Team Chairperson. The AJPO requires that the status of the E&V task be briefed to the AJPO, as well as to the three service representatives, at annual Ada tri-service reviews.

4.2 Air Force, Army, Navy

The Air Force has assumed responsibility as lead Service on the tri-service E&V Task. However, the status of the E&V Task is briefed to the AJPO and the service representatives as required at annual Ada tri-service reviews. At these reviews, each service representative has the opportunity to request additional information on the E&V Task and to recommend modifications to the proposed E&V Task planning.

4.3 E&V Team Chairperson

The Air Force Wright Aeronautical Laboratories (AFWAL) has assumed responsibility as the lead Air Force organization for the E&V Task. The E&V Team Chairperson is an AFWAL representative who is authorized to work directly with the AJPO in the execution of the E&V Task. The E&V Team Chairperson is required to brief the status of the E&V Task to

the AJPO and services as required at annual Ada tri-service reviews. The E&V Team Chairperson is responsible for providing technical direction to the E&V Team members and for coordinating all of the E&V activities.

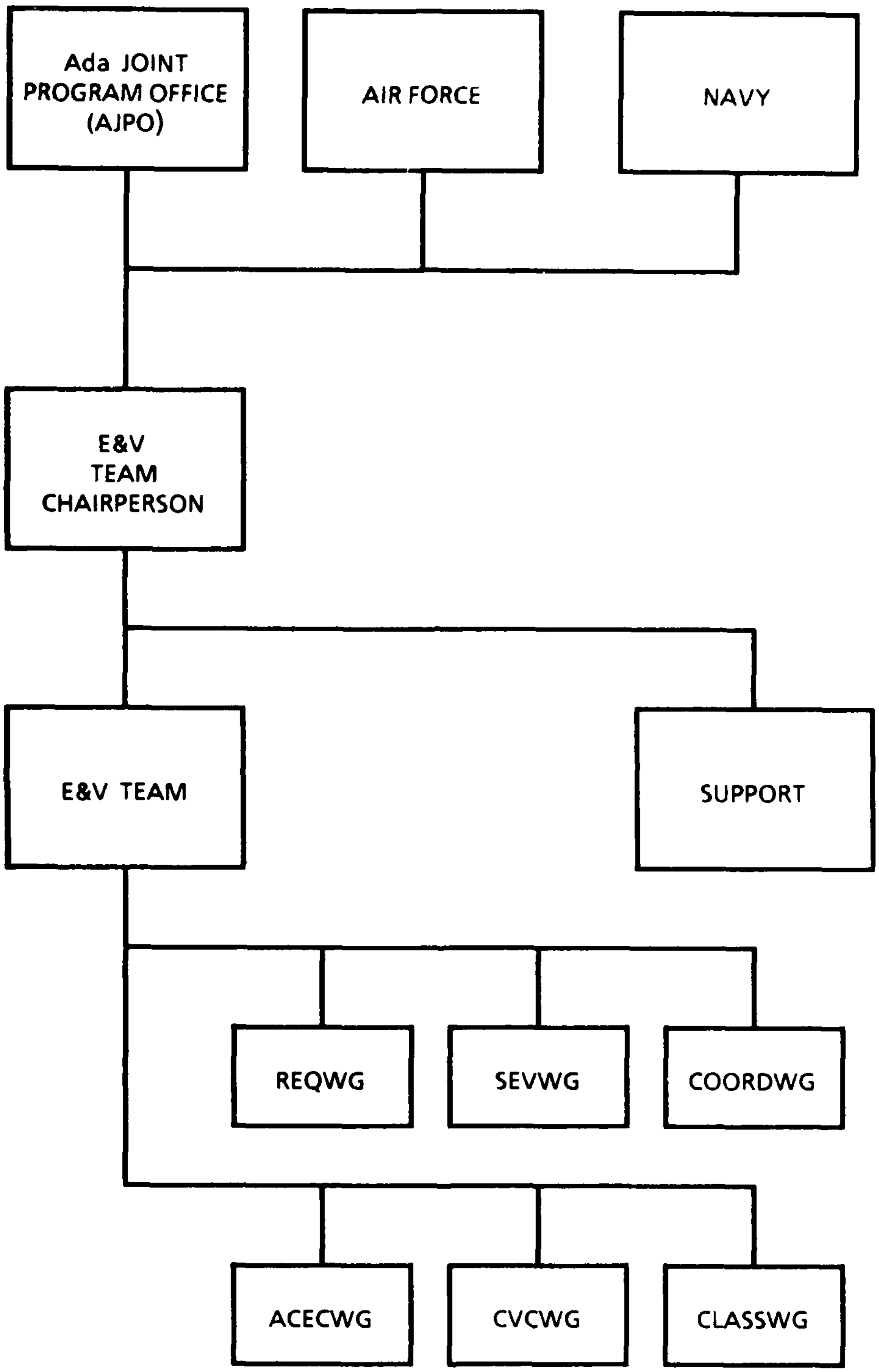

Figure A-1. E&V MANAGEMENT STRUCTURE

4.4 E&V Team

The E&V Team is composed of representatives from the following organizations:

- Air Force
  * Air Force Systems Command
  * Air Force Logistics Command
  * Air Force Communications Command
- Navy
- Other Selected Agencies
- Industry (E&V Distinguished Reviewers)
- Academia

E&V Distinguished Reviewers are those industry and academia representatives who are selected on the basis of position papers and who choose to remain actively involved in the E&V Task. These Distinguished Reviewers attend all E&V Team meetings and participate in the various working groups, They provide significant contributions to the E&V Team via their expertise and industry perspective with regard to the goals of the E&V Task.

E&V Team meetings are convened quarterly and E&V Team members are responsible for representing the technical issues/concerns of their respective organizations at these meetings. Similarly, E&V Team members are responsible for reporting the status of the E&V Team activities to their respective organizations.

4.5 E&V Team Working Groups

In order to coordinate all of the activities to be accomplished within the E&V Task, the E&V Team is partitioned into six working groups. The identification of these working groups, and their associated areas of responsibility, are delineated in the following sections. These working groups are subject to change during the life of the E&V Task. Each working group has a designated Chairperson and Vice-Chairperson. It is the responsibility of each working group Chairperson to coordinate the activities of the working group with the E&V Team Chairperson. In addition, each working group Chairperson is required to brief the status of the respective working group at every E&V Team meeting.

4.5.1 Directional Management Working Groups

4.5.1.1 E&V Requirements Working Group (REQWG)

The REQWG is responsible for performing the following tasks:

- Maintain an E&V Requirements Document against which the E&V Reference Manual will be developed.

- Provide analysis of requirements in the E&V Requirements Document to determine their adequacy, completeness, traceability, testability, consistency, and feasibility.

- Identify issues which may impact the development of E&V technology.

- Provide recommendations for acquisition of E&V tools and aids through the development of an E&V Tools and Aids Document.

- Prepare position papers through the duration of the E&V Task which address issues on E&V requirements.

4.5.1.2 E&V Standards Evaluation and Validation Working Group (SEVWG)

The SEVWG is responsible for performing the following tasks:

- Recommend specific areas of consideration for standards related to future evaluations and validations.

- Emphasize study on the CAIS.

- Review the development of the CAIS and identify areas of possible concern to E&V.

- Provide presentations to the E&V Team on the CAIS. - Provide liaison activity to the KIT.

- Prepare position papers throughout the duration of the E&V Task which address particular aspects of the CAIS as relevant to E&V.

4.5.1.3 E&V Coordination Working Group (COORDWG)

The COORDWG is responsible for performing the following tasks:

- Develop a Technical Coordination Strategy Document which will:

  * identify related technical efforts;

  * identify relationships between the E&V Task and each of the related tasks;

  * identify areas of mutual benefit to the tasks;

* identify impact of schedules;

* identify level of coordination required;

* identify issues which require resolution to the mutual benefit of the tasks involved.

- Identify professional organizations which are technically related to the E&V effort.

- Develop a Public Coordination Strategy Document which provides an approach as to how such public coordination will be performed.

- Maintain and distribute a set of E&V viewgraphs and corresponding text to allow E&V Team members to present the status of the E&V Task at public meetings.

- Prepare E&V status reports for publication in related journals and newsletters and dissemination at related conferences.

- Catalog all issues related to the E&V effort.

- Develop and maintain an E&V Project Reference List.

4.5.2 Technical Management Working Groups

4.5.2.1 E&V Ada Compiler Evaluation Capability Working Group (ACECWG)

The ACECWG is responsible for performing the following tasks:

- Provide a formal interface between the Ada community and the ACEC effort.

- Evaluate and critique aspects of the technical approach being employed on the ACEC effort.

- Evaluate and critique selected ACEC deliverables.

- Discuss and provide feedback on issues critical to the ACEC.

4.5.2.2 E&V CAIS Validation Capability Working Group (CVCWG)

The E&V CVCWG is responsible for performing the following tasks:

- Provide technical expertise to E&V chairman and team for review of CVC contractors' products and activities.

- Provide to E&V chairman and CVC project engineer recommendations regarding validation of CAIS.

- Coordinate regularly and closely with SEVWG concerning validation of DoD Standard 1838 implementations.

4.5.2.3 E&V Technology Classification Working Group (CLASSWG)

The CLASSWG is responsible for performing the following tasks:

- Serve as focal point for analysis of Reference System (Reference Manual and Guidebook).

- Solicit information and recommendations regarding E&V technology.

- Classify E&V technology.

- Aid in the technology transition of the Reference System.

- Delineate whole APSE issues.

- Recommend new areas of investigation.

4.6 Contractor Support

4.6.1 Ada Compiler Evaluation Capability

Contractor support is being used to develop an Ada Compiler Evaluation Capability (ACEC) which will enable the DoD to assess the performance characteristics of compilers.

4.6.2 CAIS Validation and Evaluation

Contractor support is being used to develop a CAIS Validation Capability (CVC) which will be used to determine conformance of an APSE to the CAIS, which has been developed by the KIT/KITIA.

Contractor support will be obtained for the purpose of developing software tools/aids to be used for evaluation of an APSE.

4.6.3 Technical Support

Contractor support is being used for the purpose of developing the E&V Reference System which consists of the E&V Classification Schema, the E&V Reference Manual, and the E&V Guidebook. The E&V Reference System is an organized collection of information on E&V technology.

5.0 E&V RELATIONSHIP TO OTHER ORGANIZATIONS

Figure A-2, page A-20, illustrates the relationship of the E&V Task to other organizations.

5.1 KIT

The purpose of the KIT, under the direction of the AJPO, is to develop a standard set of KAPSE interfaces to ensure the transportability of tools and the interoperability of data between conforming APSEs. The E&V Team will interact with the KIT for information exchange, particularly in the area of APSE interfaces, and for initial review of E&V work prior to public exposure. Several members of the E&V Team, including the E&V chairman, are also members of the KIT. The Chairperson of the KIT is also a guest member of the E&V Team.

5.2 User Groups and Professional Societies

It is anticipated that SIGAda, the Ada-JOVIAL Users Group (AdaJUG), the National Security Industrial Association (NSIA), the Electronic Industries Association (EIA), and Ada Europe will provide valuable contributions to the APSE E&V effort. The E&V Team has no formal relationship with these groups; however, the E&V Team will use some or all of these groups as regular forums for the presentation of reports and technical results of the APSE E&V effort, and will solicit inputs from members.

5.3 Standards Organizations

As with the User Groups and Professional Societies, there is no formal relationship with the Standards Organizations. However, because the E&V Task is based upon validation of KIT developed standards, the E&V Team must be familiar with the procedures for enforcement of standards. Knowledge of how standards are currently enforced will provide useful guidelines for the direction of the E&V Task.

5.4 Ada Board

The purpose of the ADA Board is to advise the director of the AJPO with regard to policy and issues related to the Ada Program. The E&V Team will interact with the ADA Board for information exchange on issues related to the APSE E&V effort. The Chairman of the E&V Team is a regular member of the Ada Board.

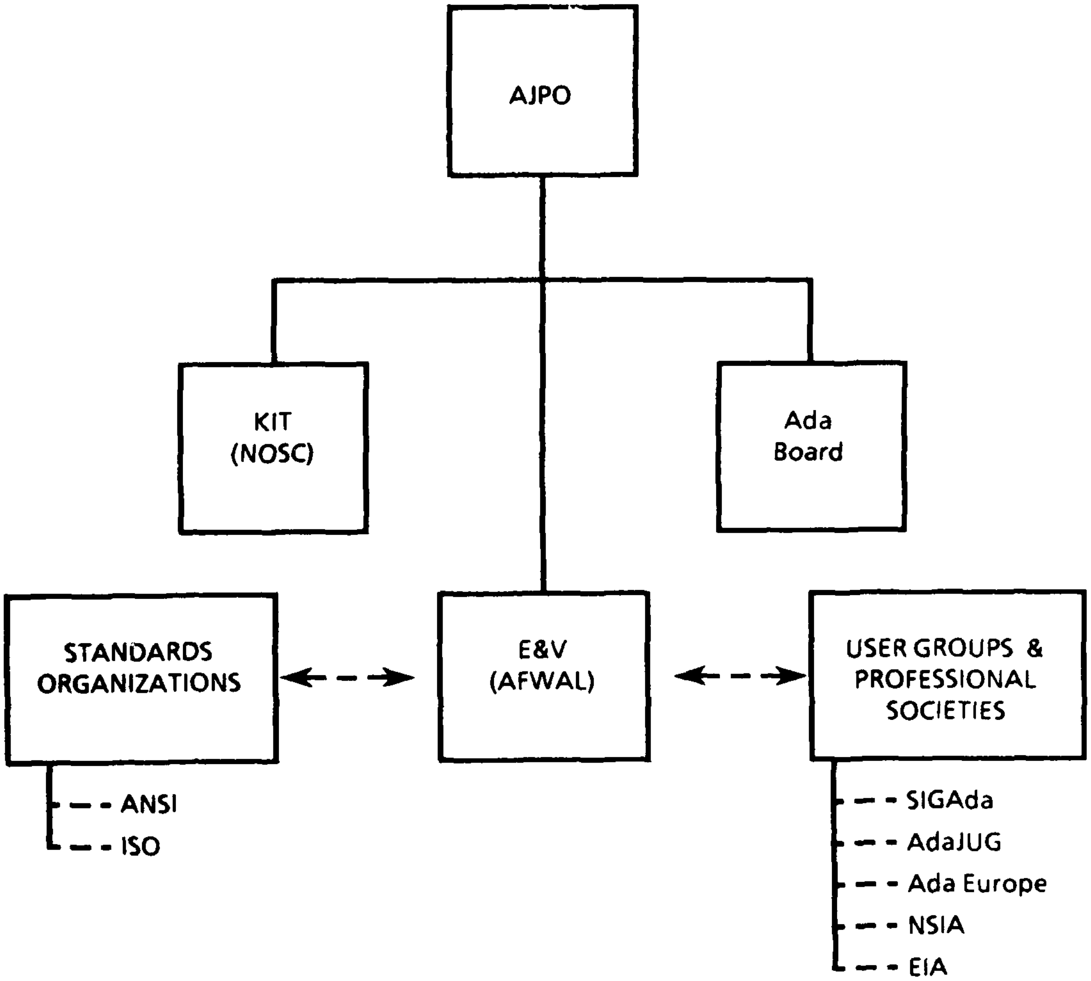

Figure A-2. E&V RELATIONSHIP TO OTHER ORGANIZATIONS

6. E&V DELIVERABLES

This section delineates each of the deliverables of the E&V Task. Working a whole, the E&V Team members, the technical consultants, and the technical support contractors, are responsible for the development of all documents. However, in order to more clearly reflect the areas of emphasis for the E&V working groups and support personnel, each document description specifies the individuals who are primarily responsible for the development of that document.

- E&V Plan

  * The E&V Plan provides a detailed and organized approach to the accomplishment of the E&V Task. The E&V Plan reflects the management approach, the technical approach and the schedules for all E&V activities. The E&V Plan is considered to be evolutionary and will be updated on an annual basis to reflect possible proposed modifications to the approach and/or schedules and to reflect accomplishments during the previous year. The E&V Team Chairperson is primarily responsible for the development of the E&V Plan.

- E&V Public Report

  * The E&V Public Report, which will be made available to the public on an annual basis, provides information on the activities of the E&V Team. The E&V Public Report contains the recorded minutes of all E&V Team meetings as well as all position papers prepared by E&V Team members. The E&V Public Report also contains E&V position papers written by industry/academia personnel seeking admittance to the E&V Team. The E&V Team Chairperson is primarily responsible for the format and collation of all entries in the E&V Public Report.

- E&V Project Reference List

  * The E&V Project Reference List provides a list of documents used as reference material by the E&V Team. Corresponding with each specified document is a synopsis which identifies the relevance of that document to the E&V Task. The E&V Project Reference List will be expanded throughout the duration of the E&V Task. The Coordination Working Group (COORDWG) is primarily responsible for the development of the E&V Project Reference List.

- E&V Technical Coordination Strategy Document

  * The E&V Technical Coordination Strategy Document identifies other ongoing DoD/contractual efforts which are technically related to the E&V Task. This document will provide a strategy for coordination between the E&V Task and each

identified effort. It will specify level of coordination, points of contact, impact of schedules of one effort on another, and benefits to be gained by each effort as a result of such coordination. This document will be updated throughout the duration of the E&V Task in order to incorporate efforts which are initiated during this time. The Coordination Working Group (COORDWG) is primarily responsible for the development of the E&V Technical Coordination Strategy Document.

- E&V Public Coordination Strategy Document

  * The E&V Public Coordination Strategy Document will identify public organizations/activities with which coordination should be established with the E&V Task for the benefit of information exchange. This document will provide a strategy for coordination between the E&V Task and each of these organizations/activities. It will specify level of coordination, points of contact, and procedures by which the plans and accomplishments of the E&V Task are presented to the organizations/activities. This document will be updated throughout the duration of the E&V Task in order to incorporate organizations/activities which are initiated during this time. The Coordination Working Group (COORDWG) is primarily responsible for the development of the E&V Public Coordination Strategy Document.

- E&V Requirements Document

  * The E&V Requirements Document will identify the requirements on E&V technology. E&V requirements will be based upon review of life-cycle methodologies, software engineering practices, human engineering aspects associated with software development, and other issues relevant to APSEs. The Requirements Working Group (REQWG) will be primarily responsible for the development of the E&V Requirements Document.

- DoD APSE Analysis

  * The DoD APSE Analysis will provide information on the features provided in the DoD APSEs. This analysis will reflect areas of commonality as well as areas of discrepancy in the manner in which functions are performed. Each revision of the DoD APSE Analysis will provide additional detail on the comparative analysis. The DoD APSE Analysis is complete.

- APSE Validation Procedures Document

  * The APSE Validation Procedures Document will provide details on the validation procedures to be implemented by organizations to which the validation execution

responsibility will be transferred. Initial versions of the APSE Validation Procedures Document will reflect general validation procedures common to existing validation organizations. Later versions of the APSE Validation Procedures Document will include APSE specific validation procedures, such as those applicable to the CAIS. The Standards Evaluation and Validation Working Group (SEVWG) is primarily responsible for the development of the APSE Validation Procedures Document.

- E&V Issues and Strategies for CAIS Document

  * The E&V Issues and Strategies for CAIS Document addresses the analysis, evaluation, and validation of the CAIS (MIL-STD 1838). Consequently, sections in this document require access to and an understanding of the CAIS. This document enumerates many of the issues and problems that should be considered for validation and evaluation of CAIS implementations, and potential solutions are presented as appropriate. This document does not provide a complete or comprehensive set of issues or solutions to these issues. The SEVWG is primarily responsible for the development of the E&V Issues and Strategies for CAIS Document.

- E&V Configuration Management Plan

  * The E&V Configuration Management Plan will specify the procedures which must be followed in order to perform Configuration Management of all E&V documents generated by the E&V Task as well as all tools/aids developed by the E&V Task. The Configuration Management Plan will be consistent with current Configuration Management policies implemented by the Avionics Laboratory at Wright-Patterson Air Force Base. The E&V Technical Support Contractor is primarily responsible for the development of the E&V Configuration Management Plan.

- E&V Classification Schema Document

  * The E&V Classification Schema Document will be used to define the approach for classification of APSE components. The initial E&V Classification Schema is provided in Section 3 (TECHNICAL APPROACH). However, as the E&V Task begins to identify and classify APSE components, the initial schema will be refined. The E&V Technical Support Contractor is primarily responsible for the development of the E&V Classification Schema Document.

- E&V Reference Manual

  * The E&V Reference Manual will provide information on the classification of APSE components. For each identified APSE component, the E&V Reference Manual will identify the

corresponding criterion/standard associated with that APSE component, as well as the metrics capability (or questionnaire entries) which are used to access that APSE component. Throughout the E&V Task, the E&V Reference Manual will be expanded to reflect finer granularity in the identification of APSE components as well as newly acquired/developed metrics capabilities. The E&V Technical Support Contractor is primarily responsible for the development of the E&V Reference Manual.

- E&V Guidebook

  * The E&V Guidebook is a companion document to the E&V Reference Manual. It provides information to the user as to how to implement the tools/techniques identified in the E&V Reference Manual for appropriate application of the E&V technology. The E&V Technical Support Contractor is primarily responsible for the development of the E&V Guidebook.

- E&V Tools and Aids Document

  * The E&V Tools and Aids Document will recommend specific E&V Tools and Aids for near term acquisition. It will also specify the rationale for establishing priorities for the acquisition of such tools and aids. The Requirements Working Group (REQWG) is primarily responsible for the development of the E&V Tools and Aids Document.

- E&V Tools and Aids

  * Based upon the E&V Tools and Aids Document, contractual efforts will be initiated for the development of such E&V tools and aids.

- CAIS Validation Capability (CVC)

  * The CAIS Validation Capability (CVC) will provide the validation capability to determine APSE conformance to the CAIS MIL-STD 1838 and the future MIL-STD 1838A. The CVC contractor is responsible for the development of the CVC.

- Ada Compiler Evaluation Capability

  * The Ada Compiler Evaluation Capability (ACEC), will consist of Test Suite, Implementor's Guide, and Test Report Reader's Guide, to enable the DoD to assess the performance characteristics of Ada compilers. The ACEC will be analogous and complementary to the existing Ada Compiler Validation Capability (ACVC), which is currently used to ensure conformance of Ada compilers to ANSI/MIL-STD-1815A. The ACEC will be both an extension and enhancement to the current ACVC test suite in that those specific compiler aspects imposed

through development of Mission Critical Computer Systems software, as defined in DoD Directive 5000.29, not presently addressed will be objectively measured and assessed. The ACEC will advance beyond compiler conformance information and will provide meaningful mission critical performance data essential for addressing the suitability of an Ada compiler for use in mission critical applications development. The ACEC contractor is responsible for the development of the ACEC.

7.0 E&V WORK BREAKDOWN STRUCTURE

Figure A-3, page A-26, depicts the areas of E&V Task responsibility which include the following:

- APSE E&V Management
- APSE E&V Requirements
- APSE E&V Reference Manual Development
- APSE Evaluation Capability
- APSE Validation Capability
- APSE E&V Tools/Aids
- APSE E&V Support

A Work Breakdown Structure (WBS) is provided for each of the above areas of responsibility.

Figure A-4, page A-27, illustrates the relationship of each WBS element to the specific objectives identified in Section 2 of this document.

APSE E&V PROGRAM

1000 - E&V MANAGEMENT
- 1100 - SYSTEMS MANAGEMENT
- 1200 - PLANNING
- 1300 - REVIEWS
- 1400 - PUBLIC COORDINATION
- 1500 - TECHNICAL COORDINATION

2000 - E&V REQUIREMENTS
- 2100 - RESOURCE REVIEW
- 2200 - REQUIREMENTS DEVELOPMENT
- 2300 - REQUIREMENTS ANALYSIS
- 2400 - SPECIAL STUDIES

3000 - E&V REFERENCE MANUAL DEVELOPMENT
- 3100 - CLASSIFICATION SCHEMA DEVELOPMENT
- 3200 - IDENTIFICATION OF APSE COMPONENTS
- 3300 - IDENTIFICATION OF CRITERIA/STANDARDS
- 3400 - IDENTIFICATION OF METRICS
- 3500 - CLASSIFICATION
- 3600 - REFERENCE MANUAL
- 3700 - MIGRATION ANALYSIS

4000 - EVALUATION CAPABILITY
- 4100 - EVALUATION CRITERIA ANALYSIS
- 4200 - EVALUATION CRITERIA DEVELOPMENT
- 4300 - DOD APSE ANALYSIS

5000 - VALIDATION CAPABILITY
- 5100 - VALIDATION ANALYSIS
- 5200 - VALIDATION PROCEDURES ANALYSIS
- 5300 - VALIDATION PROCEDURES DEVELOPMENT
- 5400 - VALIDATION DEVELOPMENT
- 5500 - VALIDATION APPLICATION

6000 - E&V TOOLS/AIDS
- 6100 - TOOLS/AIDS OBJECTIVES & REQUIREMENTS
- 6200 - TOOLS/AIDS DEVELOPMENT PLANS
- 6300 - TOOLS/AIDS DEVELOPMENT
- 6400 - TOOLS/AIDS DEVELOPMENT REVIEW
- 6500 - TOOLS/AIDS APPLICATION & ANALYSIS
- 6600 - TOOLS/AIDS MAINTENANCE
- 6700 - TOOLS/AIDS MODIFICATION
- 6800 - GUIDEBOOK

7000 - E&V SUPPORT
- 7100 - PUBLICATIONS
- 7200 - CONFIGURATION MANAGEMENT
- 7300 - DATA MANAGEMENT
- 7400 - MEETING SUPPORT

Figure A-3. APSE E&V TASK WORK BREAKDOWN STRUCTURE

WBS ELEMENT	E&V OBJECTIVES								
	0-1	0-2	0-3	0-4	0-5	0-6	0-7	0-8	0-9
1000: 1100								X	
1200								X	
1300								X	
1400									X
1500									X
2000: 2100	X			X	X		X		
2200	X								
2300	X								
2400	X								
3000: 3100		X							
3200			X						
3300			X						
3400			X						
3500			X						
3600			X						
3700			X			X			
4000: 4100				X					
4200				X					
4300				X					
5000: 5100					X				
5200							X		
5300							X		
5400					X				
5500					X				
6000: 6100						X			
6200						X			
6300					X	X			
6400						X			
6500						X			
6600						X			
6700						X			
6800						X	X		
7000: 7100	X								
7200								X	
7300								X	
7400								X	

Figure A-4. MAPPING OF WBS ELEMENTS TO E&V OBJECTIVES

7.1 1000 APSE E&V Management

- 1100 APSE E&V Systems Management

  * This WBS element provides for management of the APSE E&V Task. It further provides for a Public Report to be prepared every year. The Public Report will cover the technical accomplishments of the APSE E&V Task for the prior year and will be suitable for distribution in hard copy.

- 1200 APSE E&V Planning

  * This WBS element provides for the planning necessary to follow through and complete the APSE E&V Task. It further provides for the updating of the APSE E&V Plan on an annual basis.

- 1300 APSE E&V Reviews

  * This WBS element provides for the preparation and presentation of the E&V Task progress to the Ada Joint Program Office and the three services at the quarterly Ada tri-service Reviews.

- 1400 APSE E&V Public Coordination

  * This WBS provides for the development of a strategy by which the E&V Team will maintain coordination with the public on the progress of the E&V Task. This WBS includes preparation of E&V articles to be submitted for publication. It also includes preparation of materials which may be utilized by the E&V Team members for public presentation on the E&V Task.

- 1500 APSE E&V Technical Coordination

  * This WBS provides for the development of a strategy by which the E&V Team will maintain coordination with other related technical efforts. This WBS includes: (1) the identification of related tasks; (2) the identification of the relationships between the E&V Task and each of the related tasks; (3) the identification of areas of mutual benefit to the tasks; (4) the impact of task schedules; (5) the identification of level of coordination required; and (6) the identification of issues which require resolution to the mutual benefit of the tasks involved.

7.2 2000 APSE E&V Requirements

- 2100 APSE E&V Resource Review

  * This WBS element provides for the review of literature and documentation applicable to APSE E&V requirements. Such literature and documentation will include subjects such as

evaluation and validation studies, standards enforcement, tool functionality, APSE requirements, etc.

- 2200 APSE E&V Requirements Development

  * This WBS element provides for the development of requirements for APSE E&V. These requirements will be documented in an E&V Requirements Document which will be revised throughout the duration of the E&V Task as new requirements are identified.

- 2300 APSE E&V Requirements Analysis

  * This WBS element provides for the analysis of APSE E&V Requirements developed under WBS element 2200. This analysis will be conducted to determine completeness, traceability, testability, consistency and feasibility.

- 2400 APSE E&V Special Studies

  * This WBS element provides for any technical analysis or study not mentioned elsewhere. Specifically included are studies resulting in methods for assessing the risk associated with achieving levels of APSE E&V and cost benefit analysis that will provide a quantitative means to assist in making recommendations and decisions concerning implementation.

7.3 3000 APSE E&V Reference Manual Development

- 3100 APSE E&V Classification Schema Development

  * This WBS element provides for the development of a general schema which will be used as a basis for classification of APSE components. This schema will initially be based upon the classification schema provided in Section 3 of this document.

- 3200 Identification of APSE Components

  * This WBS element provides for the identification of APSE components, based upon the functionality and attributes presented in Section 3 of this document.

- 3300 Identification of Criteria/Standards for APSE Components

  * This WBS element provides for the identification of existing criteria or standards for each of the APSE components identified under WBS 3200. If no criteria or standards exist for a particular APSE component, then this WBS will result in recommendations for the development of criteria against which that component may be evaluated.

- 3400 Identification of Metrics for Criteria/Standards

  * This WBS element provides for the identification of existing metrics for the criteria/standards identified under WBS 3300. If no metrics exist for a particular criterion or standard, then this WBS will result in recommendations for the development of metrics associated with that criterion or standard.

- 3500 E&V Classification

  * This WBS element provides for the classification of all APSE components identified under WBS 3200, based upon the schema developed under WBS 3100 and the associated criteria/standards and metrics identified under WBS 3300 and WBS 3400, respectively.

- 3600 E&V Reference Manual

  * This WBS element provides for the documentation of the results obtained in WBS 3500 in an E&V Reference Manual.

- 3700 APSE E&V Migration Analysis

  * This WBS element provides for a continuing analysis of the results obtained under WBS 3500. One function of this WBS will be to provide recommendations for future standardization of any APSE component for which there exists a sufficient metrics capability and for which the standardization of such a component is deemed beneficial to the overall Ada program. In addition, this WBS will result in recommendations for the development of tools/aids which will provide or enhance metrics capabilities for identified APSE components.

7.4 4000 APSE Evaluation Capability

- 4100 APSE Evaluation Criteria Analysis

  * This WBS element provides for the review and analysis of existing programming environment evaluation criteria to determine applicability to the E&V Task. This WBS includes review of the Formal Qualification Tests for the existing DoD APSEs. This WBS element also includes review of ongoing standards development activities as a source for criteria development.

- 4200 APSE Evaluation Criteria Development

  * This WBS element provides for the development of evaluation criteria which will be applied to existing DoD APSEs. The evaluation criteria developed will be based upon the results of WBS elements 4100 and 4200 and will be included within the

E&V Reference Manual developed under WBS 3000.

- 4300 DoD APSE Analysis

  * This WBS element provides for the application of the evaluation criteria developed in WBS element 4300 to existing DoD APSEs. It also provides for an analysis of the features of tools available on each of the DoD APSEs to determine areas of commonality and discrepancy. This analysis will be performed in concert with an analysis of the NBS Taxonomy effort.

7.5 5000 APSE Validation Capability

- 5100 APSE Validation Analysis

  * This WBS element provides for the review and analysis of existing APSE validation studies to determine applicability to the E&V task. This WBS includes review of validation test suites, such as the ACVC and KAPSE FQT tests.

- 5200 APSE Validation Procedures Analysis

  * This WBS element provides for the review and analysis of existing validation procedures to determine applicability to the E&V Task. This WBS includes review of ACVC procedures, as well as procedures implemented by such organizations as ANSI and ISO.

- 5300 APSE Validation Procedures Development

  * This WBS element provides for the development of validation procedures to be implemented by organizations to which the validation execution responsibility will be transferred.

- 5400 APSE Validation Development

  * This WBS element provides for the development of validation procedures which will be applied to existing DoD APSEs.

- 5500 APSE Validation Application

  * This WBS element provides for the application of the validation procedures, developed in WBS 5400, to existing DoD APSEs. This WBS also provides for the analysis of results obtained from the application of the validation procedures.

7.6 6000 APSE E&V Tools/Aids

- 6100 APSE E&V Tools/Aids Objectives and Requirements

  * This WBS element provides for the identification of

objectives, teria and requirements to be used for the selection of E&V tools/aids to be acquired/developed as part of the E&V Task. These tools/aids will be used for initial evaluation and/or validation of existing DoD APSEs.

- 6200 APSE E&V Tools/Aids Development Plans

  * This WBS element provides for the analysis necessary to recommend that specific E&V tools/aids be developed. It further provides for making the recommendation, and developing plans for the development and acquisition of these tools/aids.

- 6300 APSE E&V Tools/Aids Development

  * This WBS element provides for the development and acquisition of the recommended APSE E&V tools/aids which will be used for initial evaluation and/or validation of existing DoD APSEs. This WBS includes development of the CAIS Validation Capability (CVC).

- 6400 APSE E&V Tools/Aids Development Review

  * This WBS element provides for the monitoring of the APSE E&V tools/aids development and participation in the APSE E&V tools/aids development review process. It further provides for the reporting of the results of monitoring and reviews.

- 6500 APSE E&V Tools/Aids Application and Analysis

  * This WBS element provides for the overseeing of the application of the E&V tools/aids. It further provides for the development of guidelines for the application of the tools/aids and the analyses of the results produced by their application.

- 6600 APSE E&V Tools/Aids Maintenance

  * This WBS element provides for the maintenance of the APSE E&V Tools/Aids after they are developed.

- 6700 APSE E&V Tools/Aids Modification

  * This WBS element provides for the modification of the APSE E&V Tools/Aids which may be required to correct inadequacies in the first development or to respond to changing requirements.

- 6800 Guidebook for APSE E&V Technology Application

  * This WBS provides for the development of a Guidebook for the application of the E&V technology developed in the E&V Task.

The E&V Guidebook will correspond to use of the E&V Reference Manual developed under WBS 3000. This Guidebook will be intended for public use in application to any existing support environment.

7.7 7000 APSE E&V Support

- 7100 APSE E&V Publications

  * This WBS element provides for the publication and distribution of APSE E&V requirements, policy, strategy and other applicable documents.

- 7200 APSE E&V Configuration Management

  * This WBS element provides for the Configuration Management of all APSE E&V documents generated and all tools/aids developed in the APSE E&V program.

- 7300 APSE E&V Data Management

  * This WBS element provides for the maintenance, storage and updating of all documentation and data in the APSE E&V program. It further provides for the distribution of all data in the APSE E&V program.

- 7400 APSE E&V Meeting Support

  * This WBS element provides for the technical support required in planning, preparing, conducting and reporting on formal APSE E&V meetings. These meetings are held for the purpose of establishing E&V requirements and an E&V capability.

8.0 E&V SCHEDULES/MILESTONES

8.1 E&V Milestones Accomplished

The following E&V milestones were accomplished between June 1986 and June 1987:

- E&V Technical Coordination Strategy Document Version 3.0
- E&V Public Coordination Strategy Document Version 3.0
- E&V Reference Manual Draft Version 2.0 and 3.0
- E&V Configuration Management Plan Version 1.0
- E&V Classification Schema Draft Version 2.0

8.2 E&V Milestones Scheduled

The following milestones are expected to be accomplished during the remaining calendar years as indicated:

CY87 (2 QTR) - E&V Tools and Aids Document Version 1.0
- E&V Issues and Strategies for CAIS Version 1.0
- E&V Requirements Document Version 2.0

(3 QTR) - E&V Public Report Volume III
- E&V Plan Version 4.0
- E&V Guidebook Draft Version 2.0

(4 QTR) - E&V Requirements Document Version 3.0
- E&V Classification Schema Version 1.0

CY88 (1 QTR) - E&V Public Report Volume IV
- E&V Reference Manual Version 1.0
- E&V Guidebook Version 1.0

(2 QTR) - E&V Tools and Aids Document Version 2.0
- E&V Issues and Strategies (1838A) Version 1.0

(3 QTR) - E&V Plan Version 5.0

(4 QTR) - E&V Reference Manual Version 4.0
- E&V Guidebook Version 4.0

CY89 (1 QTR) - E&V Public Report Volume V
- E&V Issues and Strategies (1838A) Version 2.0
- E&V Reference Manual Version 2.0
- E&V Guidebook Version 2.0

9.0 E&V REFERENCES

1. Department of Defense, Requirements for High Order Computer Programming Languages -- "STEELMAN", Defense Advanced Research Projects Agency, June 1978.

2. Buxton, J.N., Requirements for Ada Programming Support Environments -- "STONEMAN", U.S. Department of Defense, February 1980.

3. Houghton, R., "A Taxonomy of Tool Features for the Ada Programming Support Environment (APSE)", U.S. Department of Commerce, National Bureau of Standards, December 1982, Issued February 1983.

4. Castor, V., "Criteria for the Evaluation of ROLM Corporation's Ada Work Center", Air Force Wright Aeronautical Laboratories, Wright-Patterson AFB, January, 1983.

5. Kafura, D., Lee, J.A.N., Lindquist, T., Probert, T., "Validation in Ada Programming Support Environments", Virginia Polytechnic Institute and State University, December, 1982.

6. OUSD (R&E) Memorandum for the Commander, U.S. Army Materiel Development and Readiness Command (DRCDE-SB), Chief of Naval Material (MAT 08Y), Commander Air Force Systems Command (ALR); "Subject: Proposed Evaluation and Validation Tasking", 15 Jun 83.

7. HQ Air Force Systems Command (ALR) Letter to AJPO, "Subject: Proposed Evaluation and Validation Tasking (Your Ltr, 15 Jun 83)", 7 Oct 83.

APPENDIX A

Acronyms

ACEC . . . . Ada Compiler Evaluation Capability

ACECWG . . . . . . Ada Compiler Evaluation Capability Working Group

ACVC . . . . Ada Compiler Validation Capability

AdaJUG . . . . . . Ada-JOVIAL Users Group

AFWAL . . . . . . Air Force Wright Aeronautical Laboratories

AJPO . . . . Ada Joint Program Office

ALS . . . . Ada Language System

ALS/N . . . . . . Ada Language System/Navy

ANSI . . . . American National Standards Institute

APSE . . . . Ada Programming Support Environments

AVO . . . . Ada Validation Organization

CAIS . . . . Common APSE Interface Set

CLASSWG . . . . . Technology Classification Working Group

COORDWG . . . . . Coordination Working Group

CVC . . . . CAIS Validation Capability

CVCWG . . . . . . CAIS Validation Capability Working Group

DIANA . . . . . . Descriptive Intermediate Attributed Notation for Ada

DoD . . . . Department of Defense

E&V . . . . Evaluation and Validation

ECS . . . . Embedded Computer Systems

EIA . . . . Electronic Industries Association

FQT . . . . Formal Qualification Testing

HOLWG . . . . . . High Order Language Working Group

ISO . . . . International Standards Organization

KAPSE	Kernel Ada Programming Support Environment
KIT	KAPSE Interface Team
KITIA	KAPSE Interface Team from Industry and Academia
MOA	Memorandum of Agreement
NATO	North Atlantic Treaty Organization
NBS	National Bureau of Standards
NOSC	Naval Ocean Systems Command
NSIA	National Security Industrial Association
REQWG	Requirements Working Group
SEVWG	Standards Evaluation and Validation Working Group
SIGAda	Special Interest Group Ada
WBS	Work Breakdown Structure

APPENDIX B

ISSUES AND STRATEGIES

FOR

CAIS EVALUATION AND VALIDATION

Version 1.0

September 1987

Prepared by

Evaluation and Validation Team

Requirements Working Group

for the

Ada* Joint Program Office

The Task for the Evaluation & Validation of Ada Programming Support Environments (APSE's) is sponsored by the Ada Joint Program Office(AJPO).

*Ada is a registered trademark of the U.S. Government (Ada Joint Program Office)

TABLE OF CONTENTS

1.0 INTRODUCTION B-4
1.1 Objective B-4
1.2 Background B-4
1.3 Standards Evaluation and Validation Working Group (SEVWG) B-4
1.4 CAIS B-5
1.5 Document Backgrounds B-6

2.0 SCOPE B-6

3.0 DESCRIPTION B-6

4.0 APPLICABILITY B-7

5.0 VALIDATION ISSUES B-7
5.1 Technical Issues B-8
5.1.1 White vs Black Box Testing B-8
5.1.2 Constraints B-8
5.1.3 Input and Output B-8
5.1.4 Queues B-9
5.1.5 CAIS Explicit Exceptions B-9
5.1.6 Limits B-9
5.1.7 Interactions B-10
5.1.8 Security (access control) B-11
5.1.9 Asynchronous Processing B-12
5.2 Non-Technical Issues B-12
5.2.1 Procedures B-12
5.2.2 Maturation/Enhancement B-12
5.2.3 Coordination B-12
5.2.4 Intent B-13
5.2.5 Subset B-13
5.2.6 Supersets B-13
5.2.7 CAIS Validation Policy B-13
5.2.8 CAIS Configuration and Identification B-13

6.0 CAIS VALIDATION B-15
6.1 Component Validation Procedures Document B-15
6.2 Purpose of the Validation Capability B-15
6.3 Test Architecture B-16
6.4 Creating a Validation Test Suite through Assertions . . . B-17
6.5 Creating a Validation Test Suite from the CAIS OD. . . . B-17
6.6 Subsets of the CAIS and Validation B-18

7.0 CAIS EVALUATION B-18
7.1 Evaluation Guidelines B-19
7.2 Representative Evaluations B-20
7.3 Access Control Mechanisms B-22

APPENDIX A Acronyms B-23

APPENDIX B E&V Team Standards and Validation Working Group Membership B-24

APPENDIX C References B-25

APPENDIX D CAIS Dependencies and Testabilities B-27

1.0 INTRODUCTION

1.1 Objective

This document is intended to provide insights and guidelines for the analysis, validation, and evaluation of implementation of the Proposed CAIS (January 1985 draft of the Military Standard Common Ada Programming Support Environment Interface Set). All subsequent unqualified uses of the acronym CAIS in this document refer to the January 1985 Draft. In this document, the key issues to evaluating and validating CAIS implementations are identified. CAIS validation is motivated and characterized, and approaches to performing validation are outlined. Further, guidelines for CAIS implementation evaluation are presented, and examples of appropriate evaluators are given.

This document is a working document developed by the Standards Evaluation and Validation Working Group(SEVWG) of the APSE Evaluation and Validation (E&V) Team to facilitate the discussion of validation and evaluation as they apply to CAIS. Future versions of this document are anticipated for the resulting standard CAIS (Military Standard CAIS 1838) and for the planned revision (Military Standard CAIS 1838A.)

1.2 Background

In 1983 the AJPO formed the Task for Evaluation and validation of Ada Program support Environment (E&V Task) and a tri-service APSE E&V Team, with the Air Force designated as lead service. The overall goal of the E&V Task is to develop the techniques and tools which will provide a capability to perform assessment of APSEs and to determine conformance of APSEs to relevant standards. As the E&V technology is developed, it will be made available to the community for use by DoD organizations, industry, and academia as deemed appropriate by the respective organizations. The E&V Task will be developing technology to evaluate specific APSE components, including the CAIS. The specific components and evaluators are enumerated in the E&V Team Requirements document and they include components such as compilers, editors, command language interpreters, and debuggers.

The E&V Task is being accomplished by an E&V Team comprised of representatives from the Air Force, Army, Navy, and other DoD organizations. Because the Air Force has assumed responsibility as lead service on this effort, the majority E&V Team members are Air Force personnel. Air Force Wright Aeronautical Laboratories (AFWAL) is lead organization for the E&V Task and the E&V Team Chairperson is an AFWAL representative.

1.3 Standards Evaluation and Validation Working Group (SEVWG)

The Standards evaluation and validation Working Group(SEVWG) is chartered to provide a forum for development of guidelines for the evaluation and validation of current, proposed and future APSE related standards and their implementations. Our intent within this document is

to establish guidelines for the evaluation and validation of the CAIS implementations and to provide technical and procedural recommendations relevant to the development of a CAIS Validation Capability (CVC.)

1.4 CAIS

The scope of the CAIS includes interfaces to those services, traditionally provided by an operating system, that affect tool transportability. The proposed CAIS contains definitions for a set of Ada package specifications that will provide a standard and transportable set of interfaces to such an underlying set of services. The purpose of the CAIS is to increase APSE tool transportability through the use of these standard interfaces. The CAIS was developed by the Kernel APSE (KAPSE) Interface Team & the KAPSE Interface Team for Industry and Academia (KIT/KITIA) as a first evolutionary step towards a full-state-of-the-art interface standard.

The KAPSE Interface Team (KIT), a tri-service organization chaired by the Navy under the guidance of the AJPO, was established in late 1981 as the result of a Memorandum of Agreement signed by the Deputy Under Secretary of Defense and the Assistant Secretaries of the three services. The KAPSE Interface Team from Industry and Academia (KITIA) was established in early 1982. The KITIA consisted of volunteer representatives from industry and academia who provided technical expertise and review capability to the KIT. The objective of the KIT/KITIA was to define a standard set of Kernel Ada Programming Support Environment (KAPSE) interfaces to ensure the interoperability of data and the transportability of tools between conforming APSE's. The Common APSE Interface Set (CAIS), developed by the KIT/KITIA, provides a common kernel interface for tools requiring device, file, and process manipulation.

In addition to the KIT/KITIA's development of the CAIS, which will require the development of a validation capability to determine conformance of implementations, other efforts have contributed to the foundation of the E&V Task. One such effort was the formation of the Ada Validation Organization (AVO), under the direction of the AJPO. The AVO is responsible for the development of an Ada Compiler Validation Capability (ACVC) which is in use to determine that Ada compiler developers have consistently implemented the standard Ada language (ANSI/MIL-STD-1815a). A second effort which contributes to the E&V task is the derivation of a taxonomy for an APSE, which systematically defines tool capabilities for a full APSE. A third effort, at the Air Force Wright Aeronautical Laboratories, provided an initial APSE evaluation questionnaire that can be used as a baseline from which to develop a more comprehensive and generic set of questions. Finally, previous and current efforts, sponsored by the AJPO, at Virginia Tech and Arizona State University have addressed issues associated with validation of Ada software interfaces, such as the CAIS.

1.5 Document Backgrounds

This Document was produced by the Standards Evaluation and Validation Working Group (SEVWG) of the E&V Team. The working group is composed of a representative spectrum of the potential CAIS users and implementors from academia, government, and industry. These potential CAIS users possess a variety of different perspectives of the CAIS which include:

- Funding agencies and end user's of tools which are principally concerned with maximizing tool transportability and who are motivated by the need to obtain a reliable mechanism for encouraging and establishing the use of CAIS-based technology;

- APSE and tool developers concerned with the flexibility, efficiency, and completeness of the CAIS standard and the ease or difficulty of using it as a means of achieving enhanced flexibility; and,

- CAIS developers that are concerned with developing tests consistent with the intent of the current proposed standard, current operational definition efforts, and anticipated future CAIS enhancements.

2.0 SCOPE

This document addresses the analysis, evaluation, and validation of the CAIS (Draft Standard January 1985). Consequently, sections in this document require access to and an understanding of the CAIS. This document enumerates may of the issues and problems that should be considered for validation and evaluation of CAIS implementations, and potential solutions are presented as appropriate. This document does not provide a complete or comprehensive set of issues or solutions to these issues.

3.0 DESCRIPTION

The issues and strategies presented in this document are related to problems that should be considered throughout the lifecycle of CAIS. Complete solutions to these issues and problems are not presented in this document, but recommendations are made as to possible solutions. Many of the issues and problems presented in this document resulted from interactions with individuals responsible for the CAIS design and subsequent prototyping efforts.

The initial four chapters of this document present introductory material including some of the motivation for the E&V Team, the SEVWG and the creation of this document. CAIS validation is then presented in two separate chapters. The first chapter raises many of the problems and issues that pertain to CAIS validation. The validation issues are presented as either technical or non-technical issues. The technical issues are those that directly affect the actual validation of CAIS implementations. The non-technical issues deal with indirect topics

such as validation of subsets/supersets, validation of future CAIS standard upgrades, technical coordination efforts with other CAIS working groups, and similar topics.

The validation chapter discusses topics dealing with validating CAIS implementations. These topics include the procedures for CAIS validation, use of existing CAIS prototypes as validation aids, and other methods for creating validation test sets.

Following the validation chapters is a chapter detailing CAIS evaluation. The organization and presentation of CAIS evaluation parallels the previous discussion of CAIS validation. The topics included in the evaluation discussion include those necessary for determining the quality of a given CAIS implementation. Evaluation topics also include the issues and problems that did not fit into the realm of validation.

The appendices of this document detail items such as acronyms, SEVWG membership, references, CAIS component dependencies and testabilities.

4.0 APPLICABILITY

This document is of interest to the designers or modifiers of the CAIS standard. It also provides insights to certain problem areas for those interested in implementing the CAIS. The CAIS validation contractor will also benefit from these preliminary investigations, as will those who are developing a prototype evaluation capability for entire APSE's. The first and foremost application of this document is the communication of this information within the E&V Team itself, and between the E&V Team and directly related activities and organizations. These include;

- E&V Technical Support Contractor

- CAIS Validation Capability (CVC) Contractor

- Government funded CAIS developers.

This document is also intended as a vehicle to communicate these issues to other interested organizations, which consist primarily of government agencies and contractors considering the utilization or development of CAIS implementations or CAIS-resident toolsets.

5.0 VALIDATION ISSUES

This chapter addresses the problems and open issues associated with validation of CAIS implementations.

5.1 Technical Issues

5.1.1 White vs Black Box Testing

The most significant issue to be addressed is the question of whether the validation test suite should be predicated on the availability of source code for the CAIS and, if so, what the affect would be of non-Ada code bodies for the CAIS. Currently, we recommend that the validation test suite should be based on black-box methodology. This means that no test will be permitted in the validation test suite that requires access to the internal source code representation of a CAIS implementation. The validation test suite should not require source code instrumentation for efficiency or functionality measurements.

5.1.2 Constraints

The scope of the CAIS validation effort must include the syntax and semantics of the Ada package interfaces. This means that the CAIS interfaces must provide the range checking that is implied in the Ada package interfaces that specify the CAIS. If the CAIS implementation is written using the Ada language, then the syntax and semantics of the CAIS interfaces are provided implicitly. If the CAIS implementation is written in a language other than Ada, though, the CAIS interfaces must provide the range checking that is implied in the Ada syntax. For example, the returned value from a CAIS function call may be returned as a CAIS constrained data type. This constraint will need to be checked by the validation capability to insure that the CAIS implementation properly implements the CAIS data type constraint.

5.1.3 Input and Output

The validation test suite must insure that information is written to and read from the magnetic tape drive as specified. This will require some type of tool outside of the CAIS, because the CAIS cannot be used to check for correctness of data written by the CAIS. The opposite is also true, the CAIS must be able to read a tape that contains information in ANSI format that was written an an external tool. The validation of the CAIS Magnetic Tape interfaces will most likely be performed in two steps. Validation needs to check both writing to and reading from the tape, but validation will involve more than simply writing some information to a magnetic tape and then immediately reading it back in. There are two reasons for this. First, the data that is being written out to the tape may never physically make it to the tape. The data might only be buffered for a while, such that a write followed by a read would only be interfacing with the buffer and not the physical tape. Second, there needs to be some method of validating that the data written out is in ANSI format. If the CAIS is allowed to write out the data and read it back in, the data could be stored in a non-ANSI format and, therefore, not be readable by other ANSI standard magnetic tape readers.

For these reasons, there will probably be some interfacing necessary between the CAIS and the external environment. Reading from the magnetic tape interfaces will be the more convenient of the two validations. A single standard magnetic tape interfaces will be the more convenient of the two validations. A single standard magnetic tape with present information stored in ANSI format can be utilized for validating the CAIS magnetic tape reading operations. Writing to a magnetic tape presents a much more complex problem to validation. After writing the information to a magnetic tape, there needs to be some mechanism outside the CAIS for validating that the results on the tape are correct.

5.1.4 Queues

Another aspect of the CAIS that is going to be difficult to validate is the implementation of queues. Queues are used within the CAIS for interprocess communication. CAIS validation must insure that the information is written to the queue and is accessible as soon as the information is written. The difficulty is that in order to determine that information is immediately available requires synchronization among two different validation tasks. Therefore, some method of validating correct queue operations is necessary that does not also involve the use of queues. This suggests that validation of queues could be performed in two ways. First, there is communication between two task that are both within the same process. It is likely that task communication between the validation tasks can be achieved using the Ada rendezvous. Second, there is communication between two task that reside in separate processes. The Ada rendezvous cannot be utilized here because of the tasks residing in separate processes, so some other form of interprocess communication must be used. In both validations, the method is similar. One task needs to write to a queue and a second task needs to validate that the queued information is immediately available for reading. The SEVWG feels that the first of these alternatives should be adopted.

5.1.5 CAIS Explicit Exceptions

The proposed standard specifies a variety of exceptions that must be examined for correct usage. The scope of the CAIS validation effort must carefully address each of the exceptions that are specified in the proposed standard. Validation should include tests that exercise each exception individually to insure proper raising and handling of each exceptional condition. The issue here is based on the varied ways in which the exception definitions can be understood.

5.1.6 Limits

The scope of the CAIS validation effort must carefully address the implementation constraints/limits and fully test the boundary conditions. For each of the pragmatic limits of the CAIS, it is recommended that a limit constant be defined and included in the standard that represents the boundary limits. This limit constant would provide the CAIS validation organization with a constant form which

boundary value tests could be generated. This constant's value would be implementation defined. For example, to fully validate that a given CAIS implementation implements the allocation of process nodes properly, a constant, eg. MAXIMUM_OPEN_NODES, could be included in the standard under Pragmatic Limits. This way, the validation of PROCESS_CONTROL's INVOKE_PROCESS would be to ensure that the implementation properly allowed "at least" 127 modes open simultaneously and also properly handled up to MAXIMUM_OPEN_NODES nodes open simultaneously. This process would increase the portability of the validation suite because the validation process would not try, for instance, to invoke more nodes than were allowable for a given implementation as defined by the limit constant. Later drafts of MIL-STD-CAIS 1838 appear to have addressed this problem through a Pragmatics Package.

5.1.7 Interactions

The CAIS Validation Capability (CVC) must consider the interactions that exist in an implementation. Three types of interactions exist within a CAIS implementation:

- Interactions with the underlying operating system or host machine,
- Interactions among CAIS interfaces, and
- Interactions between APSE tools and the CAIS implementation

In general the interactions with the underlying system may be excluded from consideration for the CVC excepting those that involve the Ada translation system. For example, CAIS implementation must generate and propagate exceptions (according to the rules of Ada) to the tool calling a facility. To do this, the CAIS implementation must directly use the method/procedures defined in the runtime system unless the implementation is in Ada. Whether the implementation is in Ada or not, tools written on top of CAIS must be interfaced using an Ada translation system compatible within the CAIS. The CVC must fully examine the generation and propagation of CAIS defined exceptions. Additionally, any other interactions between the CAIS implementation and the translation system must also be examined by CVC.

Interactions within the CAIS are the interactions that take place among interfaces. For example, one interface may be implemented in terms of another (i.e., Interfaces in the ATTRIBUTES package may be implemented by using the appropriate LIST_UTILITIES interfaces). Further, interactions may be in terms of use-sequences among CAIS interfaces. For example, a tool must open a node before accessing it or a process must be spawned before it can be awaited. While both types of interactions within the CAIS exist, their effect on the CVC is limited. Interactions that define a use sequence are helpful in identifying test programs for the CVC and making them independent of a specific implementation. That is, interfaces which are part of a use sequence are needed to test each other. One routine may be needed to initialize an input context for

another. When one CAIS facility is implemented in terms of others, the number of test programs required to validate may be reduced. That is, if TO_TEST is called from GET_NODE_ATTRIBUTE then it would not be necessary to examine all the cases in GET_NODE_ATTRIBUTE that repeat tests to TO_TEXT. While this could result in a reduced number of test cases, it would also require knowledge of the implementation itself.

Interactions with tools must also be considered in creating the CVC. Certain tools have a high dependence on the implementation details of the CAIS. Many of these tools may be needed by validation test programs. Three examples are the login program, the liner and the utility to add a new user to the system.

5.1.8 Security (access control)

The wording of the CAIS specification with respect to discretionary and mandatory access control presents the worst possible situation to validating the CAIS with regard to access control. The first paragraph of CAIS section 4.4 introduces the CAIS access mechanisms stating:

- "The CAIS specifies mechanisms for discretionary and mandatory access control (see [TCSEC]). These specifications are only recommendations. Alternate discretionary or mandatory access control mechanisms can be substituted by an implementation provided that the semantics of all interfaces in Section 5 (with the exception of Section 5.1.4) are implemented as specified."

The first problem pertains to the meaning of the clause "provided that the semantics of all interfaces in Section 5..." Since discretionary and mandatory mechanisms are included in other parts of the specification aside from the excepted section (5.1.4), does this mean that those parts must be implemented strictly as specified? If so, the parameters to any interface that creates a node (structural, process, or file) must be implemented as specified, possibly conflicting with the implementation's access control mechanisms. If these parameters need not be implemented as specified, is the implementation free to alter the format and meaning of the CAIS to accommodate the selected approach? In either event, creating a validation mechanism will be complex even if access mechanisms are completely ignored by the CVC.

The second problem is that, strictly speaking, some form of access control mechanisms is required by the CAIS as worded above. The problem in generating a CVC is that the form those mechanisms are to take is at the discretion of the implementation. Consequently, to test for conformance, the CVC must accommodate (be specifically tailored to) each implementation's mechanisms. Thus the potential exists for requiring vastly different CVC's to accommodate different approaches to access control. Our recommendation is that a more complete study of the options available to the CVC be undertaken and that future revisions of the CAIS take into account the resulting validation problems introduced.

5.1.9 Asynchronous Processing

Asynchronous Processing is a problem to validation because there does not currently exist any theoretically based approach to validating concurrency unless the concurrency is transparent to the user. Consequently, we recommend that this area be investigated further by the CVC contractor in regards to requirements for future versions of the CAIS.

5.2 Non-Technical Issues

This section will address the non-technical issues that we are concerned with in regards to the CAIS validation test suite and the procedures by which validations occur.

5.2.1 Procedures

A first draft at a validation procedures document has been developed by the SEVWG. The document, currently in E&V TEAM review, is titled "APSE Components Validation Procedures" and is intended to provided a basis for validation of the CAIS as well as other current and future military standards. The procedures document covers the mechanics of doing validations, while this document addresses the technical issues related to evaluation and validation of a CAIS.

5.2.2 Maturation/Enhancement

Future upgrades to the CAIS, for example CAIS 1838A, are expected. To allow future evolution of E&V technology, CAIS upgrades should be evolutionary in nature and should provide upward compatibility between revisions. The CAIS validation test suite should, likewise, be designed to evolve and mature as our understanding of the CAIS and other standard interfaces increases.

A number of topics were explicitly deferred in the January 1985 CAIS. These topics may be addressed in CAIS 1838A and may include topics such as:

- Configuration management
- Device control and resource management
- Distributed environments
- Inter-tool interfaces

5.2.3 Coordination

The SEVWG is maintaining an ongoing interaction with the KIT, the CAIS Implementors Group, and other organizations working on CAIS related issues. This interaction is achieved by requesting SEVWG members to participate in meetings of other organizations and then reporting back to the SEVWG. Future coordination efforts will include the CVC

contractor, the CAIS 1838A contractor and the Ada Run Time Environment Working Group (ARTEWG).

5.2.4 Intent

The intent of the CAIS validation capability (CVC) is to determine the conformance of a CAIS implementation to the CAIS. The CVC is not intended to determine whether an APSE tool implementation conforms to the CAIS (that is, to determine if an APSE circumvents the CAIS to access underlying system facilities). The CVC is not intended to assess any features of a CAIS implementation that are not specified in the standard (such as evaluating the execution efficiency of CAIS functions). While it is within the charter of the E&V Team to acquire such evaluation capabilities, we recommend that they remain outside the scope of the CVC.

5.2.5 Subset

The goal of APSE tool transportability will be compromised by a policy of allowing subsets of the CAIS. Our current understanding of the AJPO policy is that there will be no authorized subsets of the CAIS.

5.2.6 Supersets

The question of supersets is, basically, the question of whether other interfaces to the O/S are permitted to co-exist with the CAIS. The superset question includes the possibility of special functionality within the standard CAIS predicated on special combinations of parameters or special circumstances. There is no way, using only 'black box' testing methodology, to verify that only the precise semantics of the CAIS are implemented. Consequently, we recommend that an issue report be developed on CAIS supersets. the report would identify the types of supersets that could be constructed. For example, supersets could be constructed by tampering with access control mechanisms, by adding parameters to interfaces, or by allowing additional values for existing parameters. The issue report should provide an analysis of the difficulty or potential for detecting each type of superset.

5.2.7 CAIS Validation Policy

The issues regarding validations, upgrades, and corrections of the CAIS are qualitatively similar to the equivalent issues being addressed for Ada compilers. We recommend that the policy set by the AJPO for the CAIS be very similar to the policy set for Ada compilers.

5.2.8 CAIS Configuration and Identification

Issues associated with the specification of host/target configurations for Ada validations are equally applicable to CAIS validations. Recent revision of the Ada compiler validation policy [CVP85] has solved many configuration-related problems such as alleviating the expense of independent validation for similar host/target/OS triples by allowing

vendors to "derive validations". However, the current approach to identifying "equivalent" configurations is not sufficient. For example, there is no formal method for distinguishing among several variations of Motorola 68000-based systems.

With CAIS validation, the issues extend beyond those for validation of Ada compilers, since both the CAIS and the CVC are Ada applications that depend on the compiler itself. While the CAIS need not be implemented in Ada (many implementations are expected to be written partly in languages other than Ada), the CAIS is an Ada interface, and must conform to the syntax and semantics of the Ada language. It seems reasonable to require that the CAIS implementation be supported by a currently validated Ada compilation system for the CAIS target run-time environment. Naturally, the CVC must also be supported by a currently validated Ada compiler. Otherwise, one could not determine whether test failures were the result of errors in the CAIS implementation or in the Ada implementation.

Therefore, we recommend that the identification of a CAIS configuration also include the identification of the validated Ada compilation system and target Ada run-time environment which supports the CAIS implementation.

Further, this information may be necessary for customization of the test suite, or interpretation of test results. This is in addition to customizations that may be required to accommodate allowable variations in CAIS implementations, such as differences in CAIS pragmatic values. For example, the Ada language places no requirements on the supported range of numeric values, and a CVC test suite using 32-bit integers would not be acceptable to all compilation systems. While this particular difference can be accommodated by appropriate use of the Ada language, there may be as-of-yet unidentified differences that might require customization of the test suite for a particular Ada compiler.

Periphral devices present another configuration-related issue for the CVC that has not been adequately addressed for Ada compiler validation. Are such devices identified as part of a CAIS configuration? Would replacing a VT240 with a Tek 4014 require a derived validation? What role may device simulators or emulators play in a CAIS implementation and/or the CVC?

We recommend that the identification of a CAIS configuration consist of the identification of each of the following:

- Target environment
  * Hardware (including peripherals)
  * Operating system
  * Ada Run-time Environment(RTE)

- Compilation environment
  * Host system hardware
  * Operating system
  * Ada compilation system

6.0 CAIS VALIDATION

This chapter will address the validation of the CAIS. The intent of validation is to test the conformance of an implementation to the proposed standard. It should include:

- An exhaustive set of tests which assure conformance with each requirement in the standard.

- A set of tests which measure the capacity of the implementation. These could be evaluation measures but are included in validation when capacities are called for in the specification. In particular, the CAIS established reasonable guidelines with respect to minimum capacities for a useful system.

- A set of tests which validate the execution-time conformance to each requirement in the standard.

- Several large test cases from various application areas which measure whether an implementation meets requirements over long sequences of input commands. An exhaustive set of small tests may not catch errors that occur over certain specific, long sequences of input commands.

6.1 Component Validation Procedures Document

The <u>APSE Component Validation Procedures</u> document has been generated by the <u>SEVWG</u> to examine the procedures to be used in validating APSE components. Thus, the mechanics of validation are not elaborated upon here. The <u>APSE Component Validation Procedures</u> are based on the <u>Ada Compiler Validation Procedures</u> since the procedures for validating a CAIS implementation are similar to those for validating a compiler.

6.2 Purpose of the Validation Capability

The purpose of the validation capability is to test conformance to the proposed standard for implementations of that standard. This should lead to consistency among implementations which will provide for greater transportability of tools. We expect the CAIS validation capability to be administered in a manner similar to the Ada compiler validation capability. The CAIS validation capability should be developed in much the same way as the compiler capability. A set of test programs should be developed that may be used to test any CAIS implementation. The tests must be constructed so as to be applicable to any number of implementations of the CAIS.

6.3 Test Architecture

The tests must be independent of CAIS implementation details. To do this, a set of highly interdependent tests must be generated. Conceptually, each test will contain three phases, initialization, testing, and examination. Initialization will build a CAIS context against which the test will execute. For example, a test to validate OPEN will require creating a node, possibly a user's top level node, with appropriate access relationships. The node may be the target of the OPEN, or the node may be used to assure that the proper exception is generated for specific error conditions. The initialization phase must be accomplished using calls to other CAIS routines. That is, initialization may not rely on any implementation details to accomplish the contest. Further, when a test results in an unexpected outcome, one cannot be sure whether the initialization phase is at fault or the interface being tested is at fault unless all routines being called for the initialization have been validated. The testing phase of each test will call the appropriate CAIS routine providing it with the intended arguments. The last phase is to determine whether the call performed the expected action. Since the validation test program may not examine the details of a CAIS implementation, it may be somewhat difficult to establish the appropriate context and to determine whether the modification to the context was proper. Further, if the result is not as expected, then it will be difficult in general to determine what aspect of the implementation is in error. This fact arises since CAIS calls are used to establish the initial context and to examine the results of the tested interface. CAIS routines can not be ordered in such a way that all routines needed in initialization and examination can be validated prior to their use. Erroneously executing test programs can result from the use of, as yet, untested CAIS routines for initialization and examination.

Several alternative methods may be used to develop the test cases used for validation. Pragmatically, the CVC can be based on a collection of test cases that are identified by initial implementations. There are, currently, several distinct prototyping efforts underway, including Mitre, Arizona State University, IBM, TRW, SofTech, and Gould. Each effort is developing its own set of test cases that are used to exercise the prototype. These tests should be made available to the contractor responsible for developing the CVC. Although the tests may be inconsistent in format and may rely on details of the prototype which developed it, the tests nonetheless identify conditions the implementors feel necessary to exercise in their prototype. We recommend that a process be established for using/adopting prototype tests as initial elements in the CAIS validation set. At least the following four elements must be addressed in that process:

1. Are the tests correct with respect to the CAIS standard?

2. Each test must be categorized according to the requirement being tested.

3. Do the tests adopted provide for consistent coverage of the requirements?

4. Can the tests be made to be implementation independent at low cost?

It is the recommendation of the SEVWG that this method be used to establish an early prototype CVC. The implementor's tests may need to be altered significantly for consistency and to remove implementation details. Although the initial CVC will be incomplete, this does provide a mechanism to arrive at an early capability.

6.4 Creating a Validation Test Suite Through Assertions.

A method that may be used in developing or expanding the CVC is to identify test cases by using a method similar to that used to develop the ACVC. In reading the specification, test cases can be identified by constructing a set of assertions derivable from the specification. The assertions may then be used to develop test programs as deemed necessary. That is, there may not be a one-to-one mapping between test programs and the assertions. Further, there may be so many assertions that one could not afford to create all the test programs indicated. A final method that can be used to create the test programs is that suggested by Lindquist and Facemire [Lin-85]. This method identifies test cases for the interfaces from a procedural version of the CAIS, called the CAIS Operational Definition(CAIS OD). The method for generating test cases produces an extensive set of test cases in the form of input/output pairs which describe initial conditions and expected results for executions of CAIS interfaces. This method is described in more detail in the following section.

6.5 Creating a Validation Test Suite from the CAIS OD

Early in the development of CAIS, critics of the interface suggested that a more rigorous explanation of the interface's semantics be generated. Realizing that a thorough axiomatic or denotational description would be lengthy to generate and that the semantics had not been sufficiently determined to allow such a description, Lindquist [Lin-84], and Srivastava [Sri-851] suggested an abstract machine description of CAIS as an intermediate step toward a formal description. Abstract machine descriptions of CAIS process control and the node model were generated to demonstrate the technique. The abstract machine used in the descriptions is Ada-based. This has led from a textual description of CAIS to an operational description. Lindquist has been funded by the E&V Team to convert the abstract machine descriptions of CAIS into an Ada-only operational version called the CAIS Operational Definition(CAIS OD). In conjunction with this effort to establish an early and complete version of the interface that is fully transportable, Facemire[Fac-84],Coleman[Col-86], and Jenkins[Jen-86] have designed, created and enhanced an initial implementation of a technique for identifying test cases directly from the Operational Definition.

The method involves symbolic execution of the Ada code in the Operational Definition to create an execution tree delineating the distinct paths through the code. From the tree, input and output pairs are generated which represent test cases that may be included in the validation capability. Each input/output pair corresponds to one execution path through the Operational Definition. The pairs may be used to develop validation test programs. The input condition specifies the information needed in the initialization phase of a test program and the output condition specifies the information needed in the examination phase.

Thus, using the CAIS OD, test cases can be identified in a semi-automatic manner. These tests will be more complete than those generated using an ad hoc technique. Further, the CAIS OD can be used as a test bed for developing the CVC. That is, as test programs are being developed, their effectiveness can be determined by running them on the CAIS OD. Any changes necessary to cause the erroneous paths of the suite programs to execute can easily be arranged by changing the Ada code in the Operational Definition.

6.6 Subsets of the CAIS and Validation

One issue that must be addressed with the CAIS is that of subsets and the implication on validation and transportability of tools. The issue is very simply that some implementations may select only a subset of the interfaces/capabilities due to budget, machine, or other constraints. Can such implementations be validated? What are the implications on transportability of tools?

Solely from the perspective of tool transportability, complete implementations and complete validation will provide the highest degree of transportability. Nonetheless, APSE tools that might otherwise not be developed may appear implemented on top of subset CAIS implementations.

7.0 CAIS EVALUATION

An evaluation capability will be developed for the CAIS as called for in the E&V Team's requirements document. The evaluation of the CAIS must include evaluation on each of the major sections within the CAIS as well as the implementation specific characteristics. Evaluation tests the performance of an implementation and how well that implementation meets various application specific requirements in addition to those specified in the standard. Evaluation should measure:

- Disk space requirements and access times,
- Memory space requirements/constraints,
- Capacities,
- Information retrieval times,

- Portability (i.e., detect host dependencies),

- Isolation and minimization of host dependencies,

- Effects on run time or simulation performance,

- Effects on debugging tools,

- Maintainability of the implementation,

- Regression by testing the implementation with software problem reports that existed in previous versions of the implementation.

Additionally, close consideration should be given to the evaluation measures to be found in the Requirements document produced by the REQWG. In this chapter, we will present an overall set of guidelines for evaluating the CAIS implementations.

Specific areas for evaluation will include:

- Node Model

- Access Control Mechanisms

- Attributes and Relations

- Process Control

- Input/Output

7.1 CAIS Evaluation Guidelines

The following paragraphs delineate CAIS specific concerns associated with evaluation. These concerns are in addition to the standard evaluation metrics specified in the Requirements document produced by the REQWG.

This section provides additional comments on the node model evaluation criteria. The first issue deals with whether performance should be measured with apriori or empirical techniques. Clearly, in most other software domains we choose static measures which remove bias due to machine differences and report on efficiencies with respect to the number of "instructions" or space requirements per input. We can't do that very well with CAIS evaluation unless we plan to get into the code of the implementation. That implies white box evaluations, something which should be avoided. (see section 5.1).

The CAIS evaluation should test performance of the CAIS implementation. This includes such factors as node access times, disk space requirements, memory requirements, the capabilities of interface tools which are used with the CAIS such as tree walking procedures, information retrieval times and other similar performance measurements.

CAIS evaluation provides measurements that go beyond testing conformance to the CAIS standard. It tests the usefulness of the CAIS implementation and should provide the CAIS user with the kind of information needed to demonstrate if the CAIS implementation will meet the user's application requirements. The CAIS evaluation should include transportability measurements. This will require tests which can detect host dependencies and whether such dependencies are isolated and minimized as much as possible without significantly degrading performance.

Certain application requirements may dictate tradeoffs; for example, access time versus space usage. CAIS evaluations will produce information which may be weighted according to application specific requirements.

The evaluation tests should detect any CAIS implementation affects on runtime performance of generated code or simulation performance. Mechanisms needed for target debugging should also be tested if the CAIS implementation affects the performance or existence of such mechanisms. A potential source for new evaluation metrics could be recommended improvements from users of specific CAIS implementations.

On the implementation dependencies such as having a node deleted during traversal of a path, the designers of the CAIS have essentially called for undefined behavior in such situations. Nonetheless, one CAIS implementation may clearly be better than another simply because of the way that it guards against such circumstances.

We should define (perhaps by example) more specifically what is meant by expensive. Possibly, this means number of instructions or CPU time. It is clearly a challenge to define the term in such a way that measurements of it have some sort of consistent meaning regardless of the type or model of computer system hosting the CAIS.

7.2 Representative Evaluations

This section contains a list of potential elements that would be evaluated by the CAIS evaluation capability. The examples are based on the CAIS node model and are not intended to represent a complete list of evaluations for the node model. Further, the other sections of CAIS would need to be addressed in developing an evaluation mechanism.

The evaluation of the Node Model needs to address the implementation dependent aspects. What one immediately addresses for evaluation are the obvious tradeoffs such as size and time. More subtle aspects of performance, such as the time at which node traversal border is fixed, should also be addressed. Node traversal may be done each time a path is processed by CAIS. Alternatively, when first encountered by CAIS, a path may be abbreviated by a short-cut to the node. Thus, the impact of node deletion/insertion in the middle of a path will be affected by the node traversal order.

Other specific suggestions for evaluation of implementations of the CAIS node model are given below and arranged as a set of questions that would, in evaluation, be turned into evaluators.

- How expensive is it to traverse pathnames on a specific implementation of the CAIS?

- In traversal, does the cost of lookup of each (relation,key) pair in the path require a relationship look up at the next node in the path. That is, we need to evaluate how effectively relationships are represented with respect to traversing them.

- How expensive is it to attach a new relationship to a node and to remove a relationship from a node? This question addresses the fact that most implementation techniques for the node model are going to have to balance the cost of searching with the cost of inserting and deleting.

- Does the implementation of the node model efficiently handle synchronization when more than one process attempts to access node / relationships / attributes simultaneously? Independent of access control mechanisms defined by the CAIS, an implementation must efficiently protect itself from accidental damage.

- What is the level of integration/compatibility with the underlying system? This question assumes that the node model is implemented on top of an existing operating system. It addresses how well the implementation fits on top of that system.

- How much of the node model (CAIS implementation as a whole) is written in Ada? Within the confines of efficiency, we want CAIS implementations to be as transportable as possible to other systems.

- How does the implementation handle the "dangling secondary relationship" problem? This does not have to be asked directly, but can be discovered by using cleverly written test programs. Again, each of the known implementation techniques has its draw backs. For example, the operational definition uses a node access table that contains a unique node number for each currently existing node. That number is never reused. The problems with this technique include the fact that the table (or some other structure) must be searched very often in manipulating nodes, relations, and attributes.

- Another problem occurs when the system crashes. How do we assure that we will indeed generate a unique number the next time we allocate a node, and now to we know that the table remains accurate, eq. if the system crashes in the middle of an indivisible operation on the table or the node naming scheme?

- What pragmatic limits are addressed by the implementation? Are those limits viewable to a tool? If so, how?

- Are the attributes implemented as CAIS Lists? eg, do nodes actually have list_utilities lists hung off of them to represent attributes (same for path attributes)? If so then is something special done to handle the predefined attributes (which will probably not be too efficient)? If not, then the attributes must use the list utilities routines to convert from its representation to list_utilities on certain calls (TO_LIST). This may not be the most efficient way to convert, but must be done since LIST_TYPE is limited private. Note that the conversion must be done the same way in the other direction.

- A similar set of questions apply to traversing nodes, searching for attributes, creating attributes, and removing attributes. These would help focus again on the tradeoffs considered important to the implementors.

- What is the efficiency of the implementation of CAIS attributes and lists? What is the speed of basic operations including traversing nodes, searching for attributes, and etc.?

7.3 Access Control Mechanisms

In evaluating access control mechanisms, it is critically important to separate evaluation from validation. Clearly, from a performance point of view, the most important question for discretionary access mechanisms is that of the cost of access checking. To examine this question, test programs have been written to examine the efficiency with which various options available may be used. For example, checking group membership requires looking at only one role but checking where necessary rights are included in resulting rights lists requires looking at multiple roles.

Under mandatory access control, the question of evaluating access rights is again important. Test programs need to exercise this as well. Additional questions would include:

- What is the selected security model?

- How can the test suite be parameterized to allow security evaluations independent of the security model?

APPENDIX A

Acronyms

ACVC	Ada Compiler Validation Capability
AJPO	Ada Joint Program Office
ASU	Arizona State University
APSE	Ada Program Support Environment
ARTEWG	Ada Run Time Environment Working Group
CAIS	Common APSE Interface Set
CAISOD	Operational Definition (of the CAIS)
CVC	CAIS Validation Capability
DBMS	Data Base Management System
E&V TEAM	Evaluation and Validation Team
I/O	Input/Output
KAPSE	Kernel APSE
KIT/KITIA	KAPSE Interface Team(Government)/KAPSE Interface Team from Industry and Academia
OS	Operating System
REQWG	Requirements Working Group of the E&V Team
RTE	Run Time Environment(For an Ada language system)
SEVWG	Standards Evaluation and Validation Working Group

APPENDIX B

E&V Team Standards Evslustion and Validation
Working Group Membership

CURRENT MEMBERS

Chairperson:	Gary McKee	Martin Marietta Corporation
Vice-chair:	Mike Mills	U.S. Air Force, ASD-AFALC
Members:	Jeff Facemire	Texas Instruments Corporation
	Tim Lindquist	Arizona State University
	John Reddan	SYSCON Corporation

FORMER MEMBERS

Members:	Kathleen Gilroy	Software Productivity Solutions Inc.
	Bud Hammons	Texas Instruments Corporation
	Nelson Estesu	U.S. Air Force, ASD-AFALC
	LT. Jim Kirkpatrick	U.S. Air Force, AFALC
	LT. Doug Olson	U.S. Air Force, HQ AFCMD
	LT. Darleen Sobota	U.S. Air Force,AFIT

APPENDIX C

References

1. Proposed MIL-STD-CAIS,January,1985

2. MIL-STD-1815A - Ada Language Reference Manual, January,1983

3. Rationale for the Design of the MIL-STD-CAIS, KIT/KITIA,Jun/85

4. DoD Requirements and Design Criteria for the Common APSE Interface Set (CAIS), KIT/KITIA, September 1985

5. Readers Guide to the CAIS (Draft), Institute for Defense Analyses, Arlington Virginia, June 1985

6. Component Validation Procedures(CVP) Document, E&V TEAM(SEVWG),Dec/85

7. [Lin-84]Lindquist, T.E.,J.L. Facemire, and D.G. Kafura,"A Specification technique for the Common APSE Interface Set", Journal of Pascal, Ada, and Modula-2, Vol 4.,Nov 5,pp 25-32, Sept/Oct 1984.

8. [Sri-85] Srivastava, C.S., and T.E. Lindquist, "An Abstract Machine Specification for the Process Node Section of the Common APSE Interface Set (CAIS)," 3rd Annual National Conference on Ada Technology, March 1985.

9. [Lin-85] Lindquist, T.E. and J.L. Facemire, "Using an Ada-Based Abstract Machine Description of CAIS to Generate Validation Tests" ACM Washington Ada Symposium, March 1985.

10. [Lin-86] Lindquist, T.E., et.al. "Applying Semantic Description Techniques to CAIS", Proc. Third IDA Workshop on Ada Verification, May 1986.

11. [Fac-84] Facemire, J.L. "A Specification Validation Technique for the Common APSE Interface Set", Masters Thesis Virginia Tech Department of Computer Science, Blacksburg VA 1984.

12. [Col-86] Coleman, K.A., "A CAIS Validation Methodology", Masters Thesis, Virginia Tech Department of Computer Science, Blacksburg VA 1986.

13. [Jen-86] Jenkins, J. "IOGEN: A System for Generating Input Output Pairs from Ada Code", Masters Thesis, Arizona State University Department of Computer Science, Tempe AZ, 1986.

14. [KNI-84][Knisely, M. "Summary of Ada Validation Issues Raised at the October 1983 AdaJUG/AdaTEC Meeting", Kaiser Electronics, 27 January 1984.

15. [CVP-85] The Proposed CAIS Validation Policy Document, as presented to the E&V Team by Maj Al Kopp 1985.

16. [TCSEC-83] Department of Defense Trusted Computer System Evaluation Criteria (TCSEC), 1983

17. [REQ-85] Requirements for Evaluation and Validation of Ada Programming Support Environments, E&V Team (REQWG)

18. [CLAS-86] E&V Classification Schema Report (DRAFT), TASC, Reading MA 1986

APPENDIX D

CAIS Dependencies and Testabilities

This appendix is intended to provide representative examples of the sort of tests that should be considered in the development of a validation suite for the CAIS. This is an example of an ad hoc/black box approach to test suite derivation. Page numbers and item numbers correspond to the CAIS version dated 31 January 1985.

5.0 Detailed Requirements / CAIS dependencies

5.1 General Node Management

5.1.1 PACKAGE Node_Definitions
dependencies => NONE

5.1.2 PACKAGE Node_Management
dependencies => Implies that a system_level_node already exists

OPEN (P.38, 5.1.2.1)
dependencies => An accessible node should already exist
Test =>
1) Use function IS_OPEN to see if the node is open=> should return a value of TRUE.
2) Try to re-OPEN while still open => should get a STATUS_ERROR exception
3) Try to OPEN a non-existent node => should get a Name_error exception if the node is not yet created.

CLOSE (P.39, 5.1.2.2)
dependencies => An accessible node should exist and be open when this test is performed
Tests =>
1) Use function IS_OPEN to see if the node is open => should return a value of FALSE
2) Try to re-OPEN after closing => should be able to reuse the old node handle
3) Try to invoke INTENT_OF => should get a STATUS_ERROR exception because the node is closed.

GET_PARENT (P.47, 5.1.2.17)
dependencies =>
Tests=>
1) If node is the System_node => should return a NAME_ERROR exception
2) Test if "parent-of-child" is same as the original node:
uses = > ITERATE, GET_NEXT, MORE, IS_SAME, and GET_PARENT,
=> should return TRUE
3) Test if "child-of-parent" is same as the original node;

uses => GET_PARENT, ITERATE, GET_NEXT, MORE, and IS_SAME => should return TRUE

DELETE_NODE (P.53, 5.1.2.21)
dependencies =>
Tests=>
1) Use function IS_OPEN with an alternate (2nd) node handle on the deleted note to see if the node is open=> should return a value of FALSE.
2) Use function IS_OBTAINABLE with an alternate (2nd) node handle on the deleted node to see if the node is obtainable = > should return FALSE.
3) Try to OPEN the deleted note=> should get a NAME exception because the node is deleted.
4) The DELETE_NODE call must fail if the node has any primary relations emanating from it(ie. to dependent nodes)=> should get a USE_ERROR exception because the node is deleted illegally.

5.2 CAIS Process Nodes - **<not addressed>**

5.3 CAIS Input/Output - **Selected topics only.**

The general thesis here is that the CAIS function IS_MOUNTED can be used to validate that the CAIS procedure MOUNT is implemented properly. First, IS_MOUNTED can be validated in isolation and without the use of the procedure MOUNT. IS_MOUNTED is supposed to return TRUE if a tape is mounted on the tape drive represented by the file identified by TAPE_DRIVE; otherwise, it returns FALSE. Now there are two possibilities:

- First the IS_MOUNTED function could be driven by some underlying hardware/OS interrupt that is sent by the actual tape device;

- Secondly, the IS_MOUNTED function could be triggered by some operator response stating that the tape had been mounted.

The first possibility above should provide true and consistent results, but the second possibility has the potential to lie. For example, the operator could "reply" that the tape had been property mounted when in fact there was no tape on the machine at all. (I think that on most operating systems, the first case above will be enforced. But on a bare machine implementation of the CAIS, I guess that the second possibility can be constructed.) Anyway, this shows that the validation of IS_MOUNTED function will need to actually require "physically" validating that the magnetic tape was "physically" mounted (especially when an "operator response" type scenario is used by IS_MOUNTED). In this manner, the implementation of IS_MOUNTED can be validated for truly proper operation. It is not sufficient to simply issue the CAIS MOUNT procedure followed by the IS_MOUNTED function to show correct operation of the IS_MOUNTED function (the operator could lie!) Now, for validating the MOUNT procedure, the IS_MOUNTED function can be utilized after calling MOUNT to check to see if a tape is in fact mounted. This

would be greatly complicated if the MOUNT procedure were allowed to return control to the calling process before the physical mount had taken place (much like a VAX SPAWN command). The validation process would need to "estimate" the amount of time needed by an operator to "physically" walk over to put a tape on the tape drive. But this cannot be the case, because the CAIS requires that the calling task be suspended during the life of a CAIS call. Thus IS_MOUNTED should return TRUE immediately after MOUNT returns control back to the calling (validating) task. Since IS_MOUNTED has been validated earlier, it can be used for validating MOUNT. It shouldn't be necessary for the validation of MOUNT to revalidate IS_MOUNT. Thus, even if an operator reply is utilized for triggering IS_MOUNT and MOUNT, the MOUNT procedure shouldn't need to recheck that a tape was "physically" mounted during the MOUNT procedure. If an implementation truly tried to "lie/fudge" a validation on MOUNT by not actually putting a tape on the machine, this would not benefit them and would only create validation errors further down the line (for example, validating that the mag. tape device properly reads/writes ANSI formats).

5.4 CAIS Utilities - CAIS Utilities defines the abstract data type LIST_TYPE which is used by other CAIS interfaces. A list is a linearly ordered set of data elements called "list items" which may be either named or unnamed (but cannot be both). The component packages of the CAIS Utilities operate on objects of LIST_TYPE, providing the capability to insert items into a list, extract items from a list, and replace or change the values of an item in a list. There are no dependencies in the use of Lists (e.g., nothing like open before access). The only requirements are that the correct calling arguments be provided, and in nearly all cases this includes one or more lists. This kind of requirement is enforced by the strong typing of Ada during compilation. The testing of the CAIS Utilities is also fairly straight-forward. For each subprogram it is required to test that the purpose has been correctly implemented, and that the proper exceptions ware raised when errors should occur. General testing should include the following items:

- Insure that named lists and unnamed lists exist and can not be mixed.

- Insure that multiple lists of different types can exist at the same time, and will correctly be kept separate.

- Verify that an empty list returns length of zero.

- Verify that the correct 'canonical external string representations' are produced for each of the value types that can be stored in a list.

- Determine what happens if too many items are inserted in a list(this is not defined by the CAIS proposed standard at this time)

-The tests listed below are intended to be indicative of the kinds of things which should be tested, and are not intended to be exhaustive.

5.4.1 Package LIST_UTILITIES

procedure COPY (P.195, 5.4.1.5)

tests:

- Make a copy of an unnamed list, and verify that the copied list is identical in content (using EXTRACT) and length (using LENGTH).

- Make a copy of a named list, and verify that the copied list is identical using IS_EQUAL.

- Make a copy of an empty list and verify that the copied list is also empty.

- Verify the independence of the copied list by making changes to the original list and then checking to make sure the copied list still contains the original values.

function IS_EQUAL (P.195, 5.4.1.5)

tests:

- Make a COPY of a named list, and verify that the copied list is identical using IS_EQUAL.

- Make a COPY of an unnamed list, and verify that the copied list is identical using IS_EQUAL.

- Make a COPY of a list and use REPLACE to alter a list item value. Verify that IS_EQUAL now return false.

- Verify that an empty list is equal to another empty list.

- Verify that a call to IS_EQUAL for two lists which contain the same list items in different order returns false. (This can be done by copying a list and then using EXTRACT to copy an item, DELETE to remove the item from the list, and INSERT to place the item back in the list at a different location.)

procedure SPLICE (P.197, 5.4.1.9)

tests:

- Splice two lists and verify that the modified list (the one into which the copy was spliced) now contains the information from the copied list.

- Verify that splicing two named lists with overlapping items results in the raising of USE_ERROR.

- Splice two lists, modify the copied list and the resulting list differently, and the verify that neither list shows the changes made to the other list.

function SET_EXTRACT (P.198, 5.4.1.11)

tests:

- Insert an unnamed item into a list, extract an item from the same position used in the insert, and verify that the inserted item is identical to the extracted item.

- Insert a named item into a list, extract an item from the same position used in the insert, and verify that the inserted item is identical to the extracted item.

- Verify that if the whole list is extracted (position => 1 and LENGTH => positive 'last) that it is identical to the original list.

procedure REPLACE (P.201, 5.4.1.17)

tests:

- Make a copy of a list and then replace one or more values in the copied version of the list. Verify that the lists are no longer equal by calling IS_EQUAL

- Make a copy of a list and then replace one or more values in the copied version with identical values from the original list. Verify that the lists are still equal by calling IS_EQUAL.

- Verify the independence of the replaced value by replacing a value in a list, then changing the replacement items value, and finally checking that the list contains the original replacement value.

- Verify that if the procedure REPLACE is called with a name to specify the item to be replaced using an unnamed list that the exception USE_ERROR is raised.

- Verify that REPLACE raises the exception SEARCH_ERROR if it is called with a non-existent named item.

procedure INSERT (P.202, 5.4.1.18)_

tests:

- Verify that the item inserted at position zero is placed at the head of the list.

- Verify the independence of the inserted value by inserting a value in a list, then changing the inserted items value, and finally checking that the list contains the original inserted value.

- Verify that if the procedure INSERT is called with a name to identify the item to be inserted using an unnamed list that the exception USE_ERROR is raised.

- Verify that a call to INSERT with a position value larger than the current length of the list results in the raising of USE_ERROR.

- Verify that a second call to INSERT with a named value for insertion results in USE_ERROR.

function POSITION_BY_VALUE (P.202, 5.4.1.19)

tests:

- Verify that the default starting and ending positions for the search work by using them to search for values stored in the first and last items on the list.

- Verify that SEARCH_ERROR is raised if the specified value is not found.

- Verify that USE_ERROR is raised if the specified START_POSITION is larger than the current length of the list.

- Verify that a call to POSITION_BY_VALUE with an empty list raises the exception USE_ERROR.

- Verify that SEARCH_ERROR is raised if the value specified is not found.

5.4.2 Package IDENTIFIER_ITEM

procedure TO_TOKEN

tests:

- Verify that a call to TO_TOKEN followed by a call to TO_TEXT results in an identifier equal to the one originally converted.

- Verify that a call to TO_TOKEN with an invalid syntax for the identifier to be converted raises USE_ERROR.

function TO_TEXT

tests:

- Verify that a call to TO_TOKEN followed by a call to TO_TEXT results in an identifier equal to the one originally converted.

5.4.3 Generic package INTEGER_ITEM & FLOAT_ITEM

<same as for subprograms of same name in prior packages>

5.4.4 Package STRING_ITEM

<same as for subprograms of same name in prior packages>

APPENDIX C

TOOLS AND AIDS DOCUMENT

Version 1.0

September 1987

Prepared by

Evaluation and Validation Team

Requirements Working Group

for the

Ada* Joint Program Office

The Task for the Evaluation & Validation of Ada* Programming Support Environments (APSE's) is sponsored by the Ada Joint Program Office(AJPO).

CONTENTS

1.0 INTRODUCTION C-3
1.1 Purpose Of The Evaluation And Validation Task C-3
1.2 Purpose Of The Tools And Aids Document C-4
1.3 Scope C-4

2.0 TYPES OF ASSESSORS C-5
2.1 Requirements And Specifications C-5
2.2 Guidelines C-5
2.3 Metrics C-6
2.4 Benchmarks, Tests, And Test Suites C-6
2.5 Questionnaires C-6
2.6 Decision Aids C-6
2.7 Monitored Experiments C-6

3.0 ASSESSOR CAPABILITIES C-6
3.1 Near-Term Assessor Acquisition Candidates C-7
3.1.1 Compilation System Evaluators C-7
3.1.1.1 Compiler Evaluators C-8
3.1.1.2 Code Generation Evaluators C-8
3.1.1.3 Program Library Evaluators C-9
3.1.1.4 Runtime Evaluators C-9
3.1.2 Target Code Generation Aids And Analysis Toolset Evaluators C-9
3.1.3 Test Systems Assessors C-9
3.1.4 CAIS (Common APSE Interface Set) Evaluation And CAIS Validation Assessors C-10
3.1.5 Ada Design Support Evaluators C-10
3.1.6 Configuration Management Support Evaluators . . . C-10
3.1.7 Distributed Systems Development And Runtime Support Evaluators C-10
3.1.8 Distributed APSE Evaluators C-10
3.1.9 "Whole APSE" Assessors C-10
3.1.10 Transportability Evaluators C-11
3.1.11 Methodology Support Evaluators C-11
3.1.12 Interoperability Evaluators C-11
3.1.13 Multilingual APSE Evaluators C-11
3.2 Long-Term Assessor Acquisition Candidates C-11
3.2.1 Software Maturity E&V C-11
3.2.2 Software Reliability E&V C-11
3.2.3 Software Maintainability E&V C-11
3.2.4 Software Reusability E&V C-12

APPENDIX A ACRONYMS C-13

APPENDIX B E&V TEAM REQUIREMENTS WORKING GROUP MEMBERSHIP C-14

1.0 INTRODUCTION

The Tools and Aids Document is the result of deliberations of the Requirements Working Group (REQWG) of the Ada Programming Support Environment (APSE) Evaluation and Validation (E&V) Team concerning technology required to evaluate and validate APSEs and their components. This document is a reflection of the APSE E&V Requirements Document and the state of current APSE tools. It also reflects views on the subject which were obtained from a number of surveys conducted among the APSE E&V Team and appropriate ARPANet-MILNet Interest Groups.

1.1 Purpose Of The Evaluation And Validation Task

The Ada community, including government, industry, and academic personnel, needs the capability to assess APSEs (Ada Programming Support Environments) and their components and to determine their conformance to applicable standards (e.g., DOD-STD-1838, the CAIS standard). The technology required to fully satisfy this need is extensive and largely unavailable; it cannot be acquired by a single government-sponsored, professional society-sponsored, or private effort. The purpose of the Evaluation and Validation (E&V) Task is to provide a focal point for addressing the need by (1) identifying and defining specific technology requirements, (2) developing selected elements of the required technology, (3) encouraging others to develop some elements, and (4) collecting information describing existing elements. This information will be made available to DoD components, other government agencies, industry, and academia.

Validation is the process of determining conformance of an APSE or APSE component to existing standards. For example, Ada compilers are currently required to undergo validation by the Ada Validation Organization (AVO) to insure conformance to the Ada language standard (MIL-STD-1815A). In the future, validation may encompass additional standards such as the Common APSE Interface Set (CAIS) developed by the KAPSE (Kernel APSE) Interface Team/Industry and Academia (KIT/KITIA).

Evaluation is the process of assessing characteristics or attributes of an APSE or APSE component for which there may or may not be standards. Examples of such attributes include usability, efficiency, and maintainability. In the absence of standards, such attributes are free to vary across different APSE implementations. Consequently, these attributes are of interest to users when selecting between APSEs because they contribute to, or detract from, overall APSE quality and suitability for different applications or methodologies. Even in cases where standards do apply to APSE components (e.g., MIL-STD-1815A and Ada compilers), evaluations will be used to supplement information gained during validation processes.

It is anticipated that the primary benefits of E&V will be to encourage the development of quality APSEs, to promote interoperability and transportability, and to provide users and developers with a uniform and comprehensive means for assessing and selecting APSE's suitable for their specific applications and methodologies.

1.2 Purpose Of The Tools And Aids Document

There exists a critical need to support the Ada community, including compiler and tool builders as well as Ada users and educators, in the selection and improvement of APSE's and APSE components. The purpose of this document is to provide pertinent information to those agencies willing and able to fund the development of E&V Technology (these agencies include, but are not limited to, the AJPO, STARS, JIAWG, Major Program Offices of the services, etc.). To this end the Tools and Aids Document identifies the community's E&V technology needs, provides definitions of those technology needs, and prioritizes them in order of their relative importance.

In order to simplify the discussion, the term "assessor" is used in this document to refer to both tools and aids for use in evaluation and/or validation. APSE component assessors are defined in Section 2 of this document, and range through guidelines, checklists, benchmarks and experimental procedures. Acquisition of assessors includes incorporation of existing capabilities into the E&V assessment set, purchase of commercial products, or development of needed technologies and implementations of these technologies for APSE component assessment.

The Tools and Aids Document provides amplification from the APSE E&V team on:

- The kinds of assessors to acquire,
- The prioritized ordering of assessor acquisition,
- The rationale for the priorities.

1.3 Scope

The APSE E&V Requirements Document identifies APSE attributes and functionality that are perceived to require evaluation and/or validation (ie., assessment). The Tools and Aids Document identifies the kinds of assessors that need to be acquired to perform the evaluation and/or validation of the functions. The document is intended to provide the AJPO and other potential sponsors with a reference for use in the allocation of resources, RFP preparations, and source selection for Tools and Aids to support the tasks of APSE E&V.

The Tools and Aids Document is a pragmatic guide to assessor acquisition base on the APSE functions available which need evaluation and/o validation, and on the technologies and implementations of these technologies available as APSE function assessors. Through its prioritization of needs, the document emphasizes near-term acquisition of assessors. The document also provides guidance for long term assessor acquisition strategies by identifying some of the assessors that require further development.

2.0 TYPES OF ASSESSORS

Assessors are the mechanisms for providing information about certain characteristics of APSE components, including functionality, performance, maturity, and the suitability of documentation.

Types of assessors include, but are not limited to, the following:

- Requirements and Specifications
- Guidelines
- Metrics
- Benchmarks, Tests, and Test Suites
- Questionnaires
- Decision Aids
- Monitored Experiments

Each assessor type may be implemented in a number of ways, such as automated tools, tests and batteries of tests, and/or manual procedures.

2.1 Requirements And Specifications

Requirements and specifications enumerate the necessary functionality, characteristics, or performance of an APSE function or tool. These may include measures that may be made quantitatively by other assessors or by judgment alone. As standards are adopted for various APSE functions, they will be included here and used as the basis for the validation of the designated functionality.

2.2 Guidelines

Guidelines provide recommendations for the use or construction of an APSE function or component. Furthermore, guidelines may describe characteristics or qualities the tool should have.

2.3 Metrics

Metrics provide quantitative data about selected characteristics of an APSE or an APSE component.

2.4 Benchmarks, Tests, And Test Suites

Benchmarks are standard tests used to measure the execution performance or acceptability of an APSE function. Benchmarks may test one specific aspect of an APSE function, or may test a number of functions. Tests and Test Suites are instruments used to measure the performance, correctness, or other characteristics of APSE functions.

2.5 Questionnaires

Questionnaires are used to gather data not easily attainable by examination of the APSE or APSE component itself. Examples of such data might include historical information, typical usage scenarios, implementation strategies, enhancement perceptions, problems reports, etc.

2.6 Decision Aids

Decision aids allow a user to assess an APSE function from a particular point of view. Decision aids may combine the results of a number of assessors, each of which is weighted based on its usefulness for the view being considered.

2.7 Monitored Experiments

Monitored experiments, based on model projects involving an aggregation of APSE functions or tools, can be performed on APSEs or APSE components to gather data in a systematic and controlled manner. These experiments can be used for both qualitative and quantitative assessments of the functionality, usability, and performance, as well as for the more informal characteristics of APSEs.

3.0 ASSESSOR CAPABILITIES

A number of APSE function assessor capabilities have been identified as being important for providing an APSE E&V capability. Recommendations for near-term assessors are found in Section 3.1 below. The premise for near term attention is that E&V capabilities can be acquired by assembling existing assessors or by developing the assessors using existing, proven technology. They are ordered by acquisition priority determined by the E&V team.

Long term E&V capabilities require additional development of technology, or the development of more detailed requirements. Some long-term evaluator capabilities are listed in Section 3.2 below. The list should not be considered exhaustive, in that a number of other specific assessors will require development.

3.1 Near-Term Assessor Acquisition Candidates

The following prioritized list of assessment capabilities is recommended. Priorities are based on the importance to the development of mission critical software, the availability of the APSE functions to be evaluated, and the technical feasibility of developing the assessor.

1. Compilation System Evaluators
2. Target Code Generation Aids and Analysis Toolset Evaluators
3. Test Systems Evaluators
4. CAIS Evaluation and Validation Assessors
5. Ada Design Support Evaluators
6. Configuration Management Support Evaluators
7. Distributed Systems Development and Runtime Support Evaluators
8. Distributed APSE Evaluators
9. "Whole APSE" Evaluators
10. Transportability Evaluators
11. Methodology Support Evaluators
12. Interoperability Evaluators
13. Multilingual APSE Evaluators

3.1.1 Compilation System Evaluators

This section includes Compiler Evaluators, Code Generation Evaluators, Program Library Systems Evaluators and Runtime Systems Evaluators.

For the purposes of this document, the compilation system is defined as those APSE components which are Ada-specific and are required for validation: the compiler, the code generator, the program library management system, and the runtime support system. While each of these components have characteristics which should be assessed

individually, the assessment of their combined functionality will be more critical to the successful development of mission critical software.

The immediate criticality of assessor development for these four compilation system components is made evident by the many large-scale projects with requirements for the use of Ada which are presently being procured or are planned for near-term procurement. These large-scale projects include the Strategic Defense Initiative, the NASA Space Station, the STARS program, Army Tactical Command and Control System, Army WIS, and the ATF, ATA and LHX programs being evaluated for common avionics systems under the auspices of the Joint Integrated Avionics Working Group (JIAWG). The successful performance of these systems depends upon the quality/extent of code generation support and execution support found in the compilation system. APSE development teams are in the process of trying to determine which products are of sufficient quality to support the development of their complex systems. Tools to assist in these evaluations are needed now.

3.1.1.1 Compiler Evaluators

Compiler evaluators provide capabilities which measure areas such as compiler performance, code and/or time optimizations, implementation of real-time embedded programming features, usability, completeness of documentation, and completeness of configuration management and control practices. The issues being probed include how "good" are the compilers, and in what ways are they good.

It is recognized that the Ada Compiler Evaluation Capability (ACEC) contract is an attempt to provide the evaluation technology required for an Ada compiler. Available funding levels have restricted the scope of that effort to something significantly less than what is actually needed, so there is an immediate need to allocate additional funds for the procurement of compiler evaluation technology which is not found in the ACEC. The current ACEC acquisition is restricted to the provision of a test suite which can measure object code execution efficiency of Ada compilation systems.

Additional urgent requirements exist for the assessment of compiler performance, real-time embedded programming features, usability, symbolic debugging support, and other aspects of compilation that cannot be directly assessed through examination of object code.

3.1.1.2 Code Generation Evaluators

The generation of efficient code for embedded targets such as MIL-STD-1750A, 68020, 80286, etc is of prime importance in the compilation system. Assessors should evaluate both target and native host

code generators for performance, efficiency, usability, modifiability, and completeness of documentation.

3.1.1.3 Program Library Evaluators

Program Library Management Evaluator Systems include evaluators to verify characteristics such as the completeness of documentation, performance, efficiency, functional capabilities, and usability of APSE supplied program library management systems, as examples.

3.1.1.4 Runtime Evaluators

Runtime evaluators are those which measure characteristics such as the performance, efficiency, and usability of the runtime system. These would also include evaluation of the completeness of documentation and configuration management and control practices of the runtime system.

Ada Runtime evaluation is needed to evaluate the performance of target runtime support systems (RTSS), typically a runtime executive and library of runtime services. Mission critical software is particularly sensitive to timing and efficiency requirements as well as the amount of code needed for RTSS. The ability to make crucial decisions about the capability of a particular Ada RTSS to meet the demands of the application often determines the success or failure of a mission critical project. Providing sound evaluators for RTSS is essential to the success of both Ada and the mission critical systems to which it is applied. Performance measures will include the required space of the run time software. An important factor in RTSS space requirements is the ability to factor out unused services to reduce the support library size.

3.1.2 Target Code Generation Aids And Analysis Toolset Evaluators

These evaluators will provide tools to evaluate host-target system cross-assemblers; host-based target linkers and loaders; host-based target system instruction-level simulators/emulators; host-based target-code symbolic debuggers; and host-based target system instrumentation interfaces which provide visibility into target processes during mission critical software execution.

3.1.3 Test Systems Assessors

These assessors will examine the ability of the APSE or APSE component to support and facilitate the planning, development, execution, evaluation and documentation of tests of mission critical software.

3.1.4 CAIS (Common APSE Interface Set) Evaluation And CAIS Validation Assessors

CAIS assessors provide measurements about how "good" the CAIS is.

The CAIS evaluation assessment capability is to be developed to assure that the implementations of the CAIS will provide acceptable performance and other characteristics not covered by validation.

CAIS validation assessors will determine if the CAIS is in conformance with the DoD Standard.

3.1.5 Ada Design Support Evaluators

These evaluators will measure the suitability and effectiveness of various software definition, specification, and design tools. This will specifically include evaluators of Ada Program Design Language (PDL) implementations and/or guidelines in the use of Ada as a PDL.

3.1.6 Configuration Management Support Evaluators

These evaluators will examine the performance, usability, and completeness of the APSE or APSE component functionality related to controlling the contents of software systems. This will include monitoring the status, preserving the integrity of released and developing versions, and controlling the effects of changes throughout the lifetime of the software system.

3.1.7 Distributed Systems Development And Runtime Support Evaluators

These evaluators will assess the ability of the APSE or APSE Components to support software development for distributed processing systems, and to provide runtime support for distributed processing systems.

3.1.8 Distributed APSE Evaluators

These evaluators will assess the ability of two or more distributed APSEs to communicate in cooperative ways in supporting the development of mission critical software at diverse geographical locations.

3.1.9 "Whole APSE" Assessors

Assessors which assess APSE macro characteristics, such as the overall performance, efficiency, usability, completeness of documentation, and configuration management and control practices of the entire APSE system.

3.1.10 Transportability Evaluators

These evaluators assess the ease with which an APSE or APSE component can be moved to other specified hosts or APSEs without change in functionality. Transportability is measured as the degree to which this relocation can be accomplished without reprogramming.

3.1.11 Methodology Support Evaluators

These evaluators assess the extent to which the APSE or APSE components support software development methodologies.

3.1.12 Interoperability Evaluators

These evaluators assess the ability of an APSE to exchange database objects and their relationships with other specified APSEs in forms usable by APSE components and user programs without conversion. Interoperability is measured as the degree to which this exchange can be accomplished without conversion.

3.1.13 Multilingual APSE Evaluators

These evaluators assess the extent to which the APSE or APSE components support the analysis/development of mission critical software where multiple source languages are involved. Multiple source language support includes the construction of Ada programs which interface to units written in other languages; and/or the support for the maintenance of files of programs not written in Ada (such as documentation); and/or support for programs written completely in languages other than Ada (e.g., existing programs written in FORTRAN, Pascal, C, LISP, etc.).

3.2 Long-Term Assessor Acquisition Candidates

Long term assessor candidates are those that require considerable technology development.

3.2.1 Software Maturity E&V

How do we recognize and measure when a piece of software is mature?

3.2.2 Software Reliability E&V

How do we determine and measure when software is reliable?

3.2.3 Software Maintainability E&V

How do we recognize and measure when software is maintainable?

3.2.4 Software Reusability E&V

How do we determine and measure when software is reusable? Evaluation and Validation of function interaction will become more important as software development environments take on additional capabilities. E&V of management methodology support and the development environment control tools will become needed in the long term.

APPENDIX A

ACRONYMS

ACEC	Ada Compiler Evaluation Capability
ACVC	Ada Compiler Validation Capability
AJPO	Ada Joint Program Office
APSE	Ada Programming Support Environment
APSEWG	APSE Analysts Working Group
AVO	Ada Validation Organization
CAIS	Common APSE Interface Set
CAISWG	CAIS Working Group
CM	Configuration Management
COORDWG	Coordination Working Group
CVC	CAIS Validation Capability
ECP	Engineering Change Proposal
E&V	Evaluation and Validation
GFE	Government Furnished Equipment
KAPSE	Kernel Ada Programming Support Environment
KIT	KAPSE Interface Team
KITIA	KAPSE Interface Team Industry/Academia
MAPSE	Minimal Ada Programming Support Environment
REQWG	Requirements Working Group
SEE	Software Engineering Environment
SEVWG	Standards Evaluation and Validation Working Group
RTS	Run Time System

APPENDIX B

E&V TEAM REQUIREMENTS WORKING GROUP MEMBERSHIP

Chairperson :	Helen Romanowsky	Rockwell International
Vice-chair:	Marlene Hazle	MITRE Corp.
Members :	Jerry Brookshire	Texas Instruments Corporation
	Mike Burlakoff	Southwest Missouri State University
	Peter Clark	TASC
	Bard Crawford	TASC
	Dan Eilers	Irvine Compiler Company
	Linda Elderhorst	NATC
	Norman Gee	McClellan AFB
	Elizabeth Kean	Griffiss AFB
	Pat Lawlis	AF Institute of Technology
	Tom Leavitt	Boeing Military Airplane Company
	Tim Lindquist	Arizona State University
	Ronnie Martin	Purdue University
	Sandi Mulholland	General Dynamics
	Vicki Rhoden	Wright-Patterson AFB
	Mary Ann Tompkins	Lockheed Corporation
	Nelson Weiderman	Software Engineering Institute

APPENDIX D

REQUIREMENTS FOR EVALUATION AND VALIDATION

OF

ADA* PROGRAMMING SUPPORT ENVIRONMENTS

Version 2.0

4 December, 1986

Prepared by

Evaluation and Validation Team

Requirements Working Group

for the

Ada* Joint Program Office

The Task for the Evaluation & Validation of Ada Programming Support Environments (APSEs) is sponsored by the Ada Joint Program Office (AJPO).

*Ada is a registered trademark of the U.S. Government (Ada Joint Program Office)

Executive Summary

The Ada community, including government, industry, and academic personnel, needs the capability to assess APSEs (Ada Programming Support Environments) and their components and to determine their conformance to applicable standards (e.g., DOD-STD-1838, the CAIS standard). The technology required to fully satisfy this need is extensive and largely unavailable; it cannot be acquired by a single government-sponsored, professional society-sponsored, or private effort. The purpose of the Evaluation and Validation (E&V) Task, which is sponsored by the Ada Joint Program Office (AJPO), is to provide a focal point for addressing the need by (1) identifying and defining specific technology requirements, (2) developing selected elements of the required technology, (3) encouraging others to develop some elements, and (4) collecting information describing existing elements. This information will be made available to DoD components, other government agencies, industry, and academia.

The purpose of this document is to set forth requirements on the E&V task. This document is intended for use by the APSE E&V Team and by the support contractor(s) in developing technology for the evaluation and validation of APSEs, however, its use in other E&V efforts is encouraged.

This document contains three categories of requirements: (1) those on the E&V Team itself, (2) those on the E&V methods and procedures, and (3) those specifying what is to be evaluated and validated.

This document does not contain requirements on APSE tools, only on the evaluation and validation of those tools.

TABLE OF CONTENTS

1.0 INTRODUCTION D-4
1.1 Purpose Of The Evaluation And Validation Task D-4
1.2 Document Purpose And Scope. D-5
1.3 Goals D-5

2.0 GENERAL REQUIREMENTS AND CRITERIA D-5
2.1 Requirements On The Evaluation And Validation Task D-5
2.2 Requirements On Evaluation And Validation Technology D-6

3.0 APPROACH D-7
3.1 Requirements On The Evaluation And Validation Team D-7
3.2 Requirements On Technology Development. D-7

4.0 REQUIRED APSE EVALUATIONS AND VALIDATIONS D-9
4.1 Introduction D-9
4.2 Attribute Definitions D-10
4.3 APSE Evaluations. D-13
4.4 Implementation Dependent Component Evaluations D-13
4.5 Functionally Dependent Component Evaluations. D-13
4.6 Quantification of Evaluation Results. D-14

5.0 QUALITY GUIDANCE FOR EVALUATION AND VALIDATION TECHNOLOGY D-14
5.1 Introduction. D-14
5.2 Quality Requirements for Evaluation and Validation Tools and Aids. D-14
5.2.1 Automated Evaluation and Validation Tools and Aids D-15
5.2.1.1 Design Requirements D-15
5.2.1.2 Documentation Requirements. D-16
5.2.1.3 Configuration Management Requirements D-16
5.2.1.4 Quality Control Requirements. D-18
5.2.2 Evaluation and Validation Aids. D-18
5.2.2.1 Design Requirements D-18
5.2.2.2 Documentation Requirements. D-19
5.2.2.3 Configuration Management Requirements D-20
5.2.2.4 Quality Control Requirements. D-20

APPENDIX A ACRONYMS D-22

APPENDIX B E&V TEAM REQUIREMENTS WORKING GROUP MEMBERSHIP D-23

APPENDIX C REFERENCES D-24

1.0 INTRODUCTION

1.1 Purpose of the Evaluation and Validation Task

The Ada community, including government, industry, and academic personnel, needs the capability to assess APSEs (Ada Programming Support Environments) and their components and to determine their conformance to applicable standards (e.g., DOD-STD-1838, the CAIS standard). The technology required to fully satisfy this need is extensive and largely unavailable; it cannot be acquired by a single government-sponsored, professional society-sponsored, or private effort. The purpose of the Evaluation and Validation (E&V) Task is to provide a focal point for addressing the need by (1) identifying and defining specific technology requirements, (2) developing selected elements of the required technology, (3) encouraging others to develop some elements, and (4) collecting information describing existing elements. This information will be made available to DoD components, other government agencies, industry, and academia.

Validation is the process of determining conformance of an APSE or APSE component to existing standards. For example, Ada compilers are currently required to undergo validation by the Ada Validation Organization (AVO) to insure conformance to the Ada language standard (MIL-STD-1815A). In the future, validation may encompass additional standards such as the Common APSE Interface Set (CAIS) developed by the KAPSE (Kernel APSE) Interface Team/Industry and Academia (KIT/KITIA).

Evaluation is the process of assessing characteristics or attributes of an APSE or APSE component for which there may or may not be standards. Examples of such attributes include usability, efficiency, and maintainability. In the absence of standards, such attributes are free to vary across different APSE implementations. Consequently, these attributes are of interest to users when selecting between APSEs because they contribute to, or detract from, overall APSE quality and suitability for different applications or methodologies. Even in cases where standards do apply to APSE components (e.g., MIL-STD-1815A and Ada compilers), evaluations will be used to supplement information gained during validation processes.

It is anticipated that the primary benefits of E&V will be to encourage the development of quality APSEs, to promote interoperability and transportability, and to provide users and developers with a uniform and comprehensive means for assessing and selecting APSE's suitable for their specific applications and methodologies.

1.2 Document Purpose and Scope

The purpose of this document is to set forth requirements on the E&V task. This document is intended for use by the APSE E&V Team and by the support contractor(s) in developing technology for the evaluation and validation of APSEs. However, its use in other E&V efforts is encouraged.

This document contains three categories of requirements. One category, contained in Section 2, consists of requirements on the E&V Team. These represent requirements against which the organization and activities of the E&V Team can be mapped. They take the form "The E&V Team will..." (e.g., "The E&V Team will develop a validation capability to determine conformance of an APSE to all applicable standards"). A second category, also contained in Section 2, consists of requirements on the E&V methods and procedures. These take the form "The E&V technology shall be..." (e.g., "The E&V technology shall be objective"). The third category, contained in Section 4, consists of requirements on what is to be evaluated and takes the form "The 'X' attribute of the 'Y' component shall be evaluated" (e.g., "The usability of the compiler shall be evaluated").

This document does not contain requirements on APSE tools, only on the evaluation and validation of those tools.

1.3 Goals

The near term goals of the E&V task are to provide a preliminary or initial set of APSE E&V requirements and a minimal set of E&V tools that can be used to assess APSEs. In addition, a feedback mechanism will be developed, by which both comments on tools and requirements, and results of applying the tools, can be submitted and disseminated.

The primary long term goal is to establish *an evolving database* of E&V technology, and results of the application of E&V technology to all available APSEs and APSE components. It is expected that this database could be used by both the potential users and the designers of APSE tools. In addition, anyone performing E&V would have a vehicle by which to make the results available to the entire community. It is anticipated that the existence of the E&V database, along with the E&V technology, would have a long term positive effect on the quality of the available Ada support software.

2.0 GENERAL REQUIREMENTS AND CRITERIA

2.1 Requirements on the Evaluation and Validation Task

(1) The E&V Task will assist DoD and industry in the development of validation capability to determine conformance to applicable APSE standards. This includes the development of tools

and aids (e.g., test sets, test scenarios, data reduction capabilities) and other means of automated support.

(2) The E&V Task will assist DoD and industry in the development of an evaluation capability to assess attributes of APSE components for which no standards exist. This includes the development of tools, aids, and other means of automated support. The E&V Team will support these task activities as appropriate.

(3) The E&V Team will generate specific requirements concerning the components and attributes to be evaluated or validated, and prioritized statements of need for E&V technology development.

(4) The E&V Team will provide an APSE E&V Classification Schema to guide the generation of specific requirements.

(5) The E&V Task will establish mechanisms for identifying and disseminating E&V information and technology to the public. The E&V team will aid in the definition of these mechanisms.

(6) The E&V Team will solicit industry and academic participation in the development of E&V technology.

(7) The E&V Team will promote community use and acceptance of E&V technology.

(8) The E&V Team will provide a focal point with respect to APSE E&V.

(9) The E&V Team will provide a knowledge base with respect to commercially available APSEs.

(10) The E&V Team will make recommendations to the Ada Joint Program Office (AJPO) on policy issues affecting the application of E&V technology.

(11) The E&V Team will establish E&V product quality guidelines and a means of evaluating E&V technology to determine and improve the validity and value of that technology.

2.2 Requirements on Evaluation and Validation Technology

In outlining requirements on the E&V technology, the following convention is adopted to distinguish between "requirements" and "criteria." Requirements, using the word "shall," distinguish themselves in that fulfillment of the requirement can be clearly observed, while this may not be true for criteria using the word "should."

(1) APSE E&V requirements and the corresponding technology shall be applicable to current APSEs (in order to yield useful short-term results), and shall evolve with future APSEs.

(2) E&V shall address individual APSE components and APSEs as a whole.

(3) E&V technology shall not be biased toward specific APSE design features or concepts.

(4) E&V technology shall be developed and specified in such a way so as to have the evaluation and validation tests be repeatable, meaning that the same results can be expected.

(5) E&V technology shall be comprehensive in that it will consider all areas of functionality of the APSEs and their components.

(6) E&V technology should provide the capability for examining application-specific attributes.

(7) E&V technology shall be tailorable to meet the needs and priorities of specific application areas and organizations.

3.0 APPROACH

This section discusses how the requirements of section 2 will be addressed.

3.1 Requirements on the Evaluation and Validation Team

The primary means of addressing the requirements outlined in Section 2.1 is through the E&V Team Working Groups.

Requirements 2.1(1) and 2.1(2) are general requirements which serve as the overall charter for the E&V Task. The Standards Evaluation and Validation Working Group (SEVWG) is currently focusing on CAIS validation, while the Requirements Working Group (REQWG) is focusing on evaluation. Requirements 2.1(3) and 2.1(4) are the responsibility of the REQWG and the E&V technical support contractor, respectively.

Requirements 2.1(5), 2.1(6), 2.1(7), and 2.1(8) are primarily the responsibility of the Coordination Working Group (COORDWG) with assistance from the team as a whole.

Requirement 2.1(9) is the responsibility of the APSE Analysts Working Group (APSEWG).

Requirements 2.1(10) and 2.1(11) were addressed by the April, 1984 E&V workshop in Airlie, Virginia, and the entire E&V Team is responsible for continued attention to these needs.

3.2 Requirements on Technology Development

Requirement 2.2(1) will be addressed through the incremental development of E&V technology. An incremental approach will be

followed in developing requirements on the E&V Team, requirements on the methods used, and requirements on what is to be evaluated. For example, the current organizing scheme for generating requirements on what is to be evaluated is contained in Section 4. This scheme takes the form of a set of component/attribute pairs in which the components represent whole APSEs, tools within an APSE, or functions performed by an APSE. Requirements for what is to be evaluated or validated will take the general form of the 'X' attribute of the 'Y' component.

With the evolution of both APSEs and E&V technology, the nature and priorities of the attributes are likely to change as will the nature of the components. As an example of a change in the priority of attributes, the ability to interface with other tools will be very important initially since a developing APSE may not include all functionality at an early stage of development. This attribute will become less important over time as more comprehensive toolsets appear. As an example of a change in the nature of the components, with increasing integration of toolsets, components such as compilers might no longer exist as separable entities.

This document only specifies requirements for E&V technology needed in the near term. The longer-term needs for E&V involve the development of capabilities that focus on the higher-level units provided by increasing levels of integration. The future iterations of the classification scheme which serves to drive the generation of requirements for E&V will focus more on these higher-level units. Additional areas of focus for the intermediate and longer term include the following:

- Evaluation of protocols used by functional components.

- Evaluation of "CAIS-conformance."

- Evaluation of extension to scope of APSE functions as simulation/support for Ada-based program description and requirements statement languages.

- Development of new procedures and metrics for evaluation.

- Use of E&V early in the APSE development life cycle.

- Evaluation of APSEs with respect to new standards.

- Increased emphasis on evaluating the APSE as a whole rather than the individual components (Requirement 2.2(2)).

- Development of the capability to E&V project - specific, application - specific, methodology - specific APSEs (Requirement 2.2(7)).

4.0 REQUIRED APSE EVALUATIONS AND VALIDATIONS

4.1 Introduction

This section levies the requirements for developing and organizing the specific E&V tools and aids or evaluators which will be applied against the APSEs to be evaluated. As viewed in this section, components of APSEs may be tools, features of tools, sets of tools, user-viewable functions performed by the APSE, facilities or functions provided by the APSE and used by some other component or external tool, or any software which provides one or more of the four interface classes defined by the E&V Plan, Version 2.0 [1]. The evaluators themselves can be checklists, guidelines, tests, benchmarks, semi-automated tools or fully automated tools.

This section specifies requirements for tools and aids providing E&V capabilities both for assessing the functions which can be performed using an APSE or part of an APSE, and for assessing the implementations of APSE components themselves as software products. The first category of evaluators are called "functionally dependent," while the second category are called "implementation dependent." For the past several years, there has been a trend away from traditional software tools in which each tool implements exactly one function. For example, a single tool might perform the functions of compilation and editing or, conversely, in many newer environments there is the capability to compose several tools to perform a single function. Thus, since functions and tools are not in one-to-one correspondence, the tools and aids requirements for functional evaluations and validations are treated separately from those for implementation dependent evaluations and validations. Additionally, this section also specifies requirements for evaluators of "whole" APSEs, resulting in a total of three major areas of evaluators. Each of the areas is defined as a matrix with "components" as one dimension and "attributes" as the second dimension.

In order to specify the requirements for the functionally dependent evaluators, a taxonomy of APSE functions was needed. The "Taxonomy of Tool Features for a Life Cycle Software Engineering Environment" [2], commonly called the SEE taxonomy, was selected as the baseline for the E&V functional taxonomy. The E&V taxonomy, which is to be developed as part of the E&V Classification Schema, is an extension of the SEE taxonomy.

The requirement for each evaluator will be to provide a capability to assess a component/attribute pair. The requirement levied by this document will then be that tools and aids evaluating the subject attribute of the subject component shall be developed as part of the E&V evaluation capability. The attributes which are used in the requirements are defined in the next subsection. Of the E&V attributes, some apply only to the functionally dependent tool features (i.e., the taxonomy), some to the implementation dependent tool features or the APSEs as a whole, or to some combination

thereof. While it is possible that for some component/attribute pairs there will be no applicable evaluations to be performed (in the case where the attribute does not apply to the component), it is equally likely that for a given component/attribute pair there could be a need for several distinct evaluators in order to completely perform the necessary evaluation.

In addition to stating the requirements for developing the tools and aids, this section also levies the requirement to develop a methodology for quantifying and interpreting the results obtained from applying the tools and aids.

4.2 Attribute Definitions

Evaluation of an APSE component is made with respect to attributes that the component is to possess. To provide a consistent meaning, the following attribute definitions and interpretations have been adopted for E&V use.

(1) Availability - The probability that a component will be functionally ready or operable at some specified point in time. [3]

(2) Capacity - The upper or lower limit of a component's functions or features.

(3) Completeness - The extent to which a component provides the entire set of operations necessary to perform a function.

(4) Configuration Management Practices Applied -The provision of activities related to controlling the contents of a component; including monitoring the status, preserving the integrity of released and developing versions, and controlling the effects of changes throughout the component. [3]

(5) Correctness - Agreement between a component's total response and the stated response in the functional specification (functional correctness), and/or between the component as coded and the programming specification (algorithmic correctness).

(6) Costs - The cost of a complete component or the costs of features of a component. The cost of a component may vary depending on delivery with source code or object code only (for example). Other cost considerations are installation, user assistance, and maintenance support.

(7) Documentation- The technical data, including on-line, documentation, listings, and printouts, which serve the purposes of: (1) elaborating the design or details of a component, (2) explaining the capabilities of a component, and (3) providing operating instructions for using the component to obtain desired results. [3]

(8) Efficiency - The extent to which a component fulfills its purpose using a minimum of computing resources. This implies that choices of source code constructions are made in order to produce the minimum number of words of object code, or that where alternate algorithms are available, those taking the least time are chosen; or that information-packing density in core is high etc. [3]

(9) Extendability (Extensibility) - The extent to which a component allows new capabilities to be added and existing capabilities to be easily tailored to user needs. [3]

(10) Granularity - The degree to which a component has separate limited functions that are composable, user selectable, and communicate through a common data base.

(11) Hardware Dependence - The design and implementation features of a component that take advantage of host or target hardware techniques and performance.

(12) Integrity - The extent to which access to a component or associated data by unauthorized persons can be controlled.

(13) Interface Characteristics - The set of assumptions made by the component and made about the component by the remaining components and the system in which it appears. Software components have control, data, and service interfaces. Included in this attribute is conformance to any existing pertinent interface standards. [3]

(14) Interoperability - The ability of APSEs to exchange data base objects and their relationships in forms usable by components and user programs without conversion. Interoperability is measured by the degree to which this exchange can be accomplished without conversion.

(15) Intraoperability - The ability of APSE components to exchange data base objects and their relationships in forms usable without conversion.

(16) Maintainability - The extent to which a component facilitates updating to satisfy new requirements or to correct deficiencies. This implies that the component is understandable, testable, and modifiable. [3]

(17) Maturity - The extent to which a component has been used in the development of deliverable software by typical users and to which the feedback from that use has been reflected in modifications to the component.

(18) Transportability - The ability of a component to be installed on a different APSE without change in

functionality. Transportability is measured in the degree to which this installation can be accomplished without reprogramming.

(19) Power - The extent to which a component has capabilities, such as default options and wild card operations, that contribute to the effectiveness of the user.

(20) Proprietary Rights - Restrictions on the release, distribution, or use of a component. This includes so called "data rights" restrictions.

(21) Rehostability - The ability of an APSE component to be installed on a different host or different operating system with needed reprogramming localized to the KAPSE or machine dependencies.

(22) Reliability - The extent to which a component can be expected to perform its intended functions in a satisfactory manner. [3]

(23) Resources Required - The amount and types of hardware or software facilities needed for the operation of a component. This includes primary and secondary storage and any other required resources.

(24) Retargetability - The ease with which an APSE component can accomplish its function with respect to another target. The component may require modification.

(25) Robustness - The protection of a component from itself, user errors, and system errors. The ability to recover and provide meaningful diagnostics in the event of unforeseen situations.

(26) Software Production Vehicle - The methodology(ies), language(s), and technique(s) used to produce the software related to a component.

(27) Test Availability - The availability of tests that verify the correctness or effectiveness of a component function or feature. These tests may also verify proper response for an incorrect input or technique.

(28) Testability - The extent to which a component facilitates the establishment of verification criteria and supports evaluation of its performance. This implies that requirements are matched to specific modules, or diagnostic capabilities are provided, etc. [3]

(29) Usability - User effort required to learn, operate, prepare input for, and interpret output of a component.

4.3 APSE Evaluations

This section levies the requirement for developing the tools and aids needed to macroscopically evaluate an entire APSE as an integrated tool or, in other words, evaluators to assess "whole" APSE issues. For example, the "APSE/hardware dependencies" pair would specify a requirement for an evaluator which would assess the hardware types, peripheral types, and configurations needed to implement the subject APSEs.

Requirement:

A set of tools and aids shall be developed for evaluating Ada Programming Support Environments with respect to the E&V attributes.

Requirement:

Procedures shall be developed that specify how to apply E&V technology.

4.4 Implementation Dependent Component Evaluations

This section levies the requirements on implementation dependent component evaluations. The evaluators specified by these requirements are used to assess the quality of APSE components (i.e. APSE tools) as pieces of software independent of the functions performed by the components. As a consequence, it is expected that, to a great extent, the same set of tools and aids can be applied to all APSE components uniformly rather than requiring a separate set for each type of component. As a result of this uniformity and the fact that different APSEs can consist of incomparable sets of tools, the actual APSE components are not specified in this document. An example of an evaluator resulting from this requirement is an evaluator which can be used to assess the maintainability of an arbitrary APSE tool.

Requirement:

A set of tools and aids shall be developed for evaluating individual Ada Programming Support Environment components with respect to the E&V attributes.

4.5 Functionally Dependent Component Evaluations

This section levies the requirement to produce evaluators of the functionality of APSEs and their components. As discussed earlier, the E&V taxonomy, which is based on the SEE taxonomy, is used to specify the requirements. The attributes to be applied to each function and subfunction described in the taxonomy are used to evaluate how a particular function is performed in the APSE, rather than evaluating the piece of software implementing the function as in the previous section. In the whole APSE and APSE component evaluations, it is

expected that a single evaluator will apply to a large class of components. In contrast, it is expected that, in many cases, a functionally dependent evaluator will apply only to a specific component/attribute pair.

Requirement:

A set of tools and aids shall be developed for evaluating each of the Ada Programming Support Environment functions specified in the E&V taxonomy with respect to the E&V attributes.

4.6 Quantification of Evaluation Results

This section contains requirements for developing a methodology and appropriate tools and aids for quantifying, recording, and interpreting the results obtained by the application of the tools and aids specified by the requirements stated in the three previous sections.

Requirement:

A methodology for quantifying and recording the results of applying the E&V tools and aids shall be developed, implemented, and documented.

Requirement:

Guidelines, tools, and aids for interpreting the results of applying the E&V tools and aids shall be developed, implemented, and documented.

5.0 QUALITY GUIDANCE FOR EVALUATION AND VALIDATION TECHNOLOGY

5.1 Introduction

Requirement 2.1(11) states that "The E&V Team will establish E&V product quality guidelines." This section establishes the requirements for detailed guidelines to be developed. Quality is defined to be the total composite of product characteristics through which the product will meet the expectations of the user.

5.2 Quality Requirements for Evaluation and Validation Tools and Aids

Specific instances of E&V technology can be thought of as lying within a spectrum that varies in terms of levels of automation. At one extreme are those capabilities that are only automated to the extent of providing on-line files that can be copied and edited to include information gathered as a result of manual E&V activities. At the other extreme are those capabilities that are totally automated to the extent that, when an APSE component is specified for an evaluation or validation, the results are obtained with very little human intervention. For the remainder of this section, the former capabilities will be referred to as E&V aids; the latter will

be referred to as automated E&V tools. The following presents the quality requirements for automated E&V tools and E&V aids, respectively.

5.2.1 Automated Evaluation and Validation Tools and Aids

Examples of automated E&V tools may include test scenarios, test sets, and data reduction capabilities; or automated static analyzers and dynamic analyzers implemented to support metrics of interest to E&V. This form of E&V technology is characterized by the medium used to provide primary capabilities, i.e., software. Accordingly, the design, documentation, configuration management, and quality control requirements stated below have been formulated with emphasis on characteristics associated with software products.

5.2.1.1 Design Requirements

Automated E&V tools shall be designed to satisfy required APSE evaluations and validations as specified in Section 4.0 and the requirements on E&V methodology as specified in Section 2.2 of this document. In addition,

- Automated E&V tools shall be designed to provide capabilities that are directly traceable to E&V Requirements as specified and elaborated in Section 4.0 of this document, the E&V Reference Manual, and the E&V Guidebook.

- Automated E&V tools shall be designed to satisfy predetermined requirements and thresholds associated with applicable attributes as defined in Section 4.2.

- Automated E&V tools shall be designed such that they are generally applicable to APSEs, rather than be dependent upon features of a specific APSE.

- Automated E&V tools should be designed to be self-contained.

- Automated E&V tools shall be designed to support the self-checking of results.

- Automated E&V tools shall be designed to provide a consistent user interface.

- Automated E&V tools should be designed to support objective evaluations as opposed to subjective evaluations.

- Automated E&V tools should be designed to allow execution of groups of tools with minimum user interaction.

5.2.1.2 Documentation Requirements

Each automated E&V tool shall be accompanied by a user's manual that defines, as a minimum:

- The purpose of the tool.
- Required hardware/software configuration(s).
- Initialization procedures for files, variables, and other parameters, as needed.
- Execution options available to the user.
- User inputs including format, frequency, allowable range, and units of measure.
- Step-by-step procedures for execution.
- Termination procedures.
- Restart procedures.
- Expected outputs including format, frequency, allowable range, units of measure.
- Procedures for interpreting results.
- Error messages.
- Diagnostic features.

5.2.1.3 Configuration Management Requirements

E&V sponsored automated E&V tools shall be placed under configuration management procedures which (1) identify and document the functional and physical characteristics of each automated E&V tool, (2) control changes to those characteristics, and (3) record and report the processing of changes and the status of implementation. Required configuration management activities include:

- Configuration identification that indicates the relationship between the automated E&V tools and their documentation. This includes identifying the documentation that establishes and defines the functional and allocated baselines, and the product baseline; and identifying all documentation and computer software media containing code, documentation, or both by titling, labeling, numbering, and cataloging procedures. These procedures shall uniquely identify the specific versions of each automated E&V tool to which a document applies; the serial, edition, change status, and other identification details of each

document; and, the specific contents of each medium, including change status.

- Configuration control to control all changes to the formally baselined documents, and code for each automated E&V tool. This includes the implementation of a corrective action system to report and track all problems and to implement necessary changes.

- Configuration status accounting to generate periodic status reports on all products in the allocated and product baselines. Status reports shall: (1) provide traceability of changes to controlled products, (2) serve as a basis for communicating the status of configuration identifications and associated software, and (3) serve as a vehicle for ensuring that delivered documents describe and represent the associated software.

Version Description Documents shall be prepared to identify new versions and associated documentation for each automated E&V tool. Each Version Description Document shall include, as a minimum:

- Inventory of materials released, including a list of physical media and associated documentation which make up the new version of the automated E&V tool.

- Inventory of automated E&V tool contents identifying all software that is being released by reference to appropriate specifications and manuals and by listings.

- List of all changes installed since the previous version/change with a cross reference to the affected specifications. The ECP number and date, and the related software change number/change package and date, shall also be indicated for each entry in the list.

- Interface compatibility indicating other automated E&V tools that are affected by the changes incorporated in this release.

- Bibliography of reference documents listing all documents pertinent to the initial release of the automated E&V tool and identifying any changes to the listed documents.

- Description of the operational effect of each ECP implemented in this version.

- Possible problems and known errors and steps being taken to resolve them.

5.2.1.4 Quality Control Requirements

Each automated E&V tool shall be developed in accordance with pre-determined requirements. In addition, procedures shall be in place to assess conformance to the requirements, take corrective action when necessary, and plan for improvements in both the product and the process.

Each automated E&V tool shall be tested throughout its development in accordance with an appropriate software testing methodology. For example, formal validations shall be tested in accordance with an extremely rigorous testing methodology. Subjective evaluations do not require the same degree of thoroughness during the testing process.

When feasible, each automated E&V tool shall undergo formal acceptance testing that demonstrates the ability of the tool to satisfy all specified functional and quality requirements. When it is not feasible to formally demonstrate the satisfaction of requirements in this manner, sufficient evidence resulting from the appropriate analyses shall be made available to indicate the tool's ability to satisfy requirements.

To ensure the suitability of the tool for the intended end-user, each automated E&V tool shall undergo beta testing conducted by a set of users that represent the spectrum of expected end-users. The purpose of this testing activity is to allow the determination of the suitability and effectiveness of the automated E&V tool in the operational environment.

Finally, each automated E&V tool should undergo testing and evaluation by an independent group of individuals who are experts with respect to the objectives of, and the procedures for, the evaluation and validation of APSEs.

5.2.2 Evaluation and Validation Aids

Examples of E&V aids include questionnaires, checklists, and manual decision aids. This form of E&V technology is characterized by the degree to which manual activities are required to carry out the evaluation/validation process. Thus, the human engineering qualities of the aids with respect to the intended users are of utmost importance. It should be noted, however, that some degree of automation may be available to support E&V aids and thus software-related quality concerns are also addressed in the following.

5.2.2.1 Design Requirements

E&V aids shall be designed to satisfy required APSE evaluations and validations as specified in Section 4.0 and the requirements on E&V methodology as specified in Section 2.2 of this document. In addition,

- E&V aids shall be designed to provide capabilities that are directly traceable to E&V requirements as specified and elaborated in Section 4.0 of this document, the E&V Reference Manual, and the E&V Guidebook.

- E&V aids shall be designed to satisfy pre-determined requirements and thresholds associated with applicable attributes as defined in Section 4.2 of this document.

- Specifications for individual E&V aids shall include requirements and thresholds pertaining to both these attributes and those of Section 5.2.1.1.

- E&V aids shall be designed such that they are applicable to generic APSEs.

- E&V aids shall be designed to be self contained and independent.

- When feasible, E&V aids shall be designed to support the self-checking of results.

- E&V aids shall be designed to provide a consistent user interface, as appropriate, depending on the expected class of user (e.g., management, technical).

- When feasible, E&V aids shall be designed to support objective evaluations as opposed to subjective evaluations.

- Designs for E&V aids shall be reviewed for technical adequacy, compatibility and consistency with prior products, comprehensiveness, understandability, and ability to satisfy required quality attribute thresholds.

5.2.2.2 Documentation Requirements

Each E&V aid shall be accompanied by a User's Manual that defines, as a minimum:

- The purpose of the aid.

- Intended users and level of expertise required.

- Application options available to the user.

- User input including format, frequency, allowable range, and units of measure.

- Step-by-step procedures for application.

- Procedures for interpreting results.

In addition, for automated portions of each E&V aid the information described in Section 5.2.1.2 must also be included.

5.2.2.3 Configuration Management Requirements

E&V Task sponsored E&V aids shall be placed under configuration management procedures which provide technical and administrative direction and surveillance to: (1) identify and document the functional and physical characteristics of each E&V aid, (2) control changes to those characteristics, and (3) record and report the processing of changes and the status of implementation. Required configuration management activities include those listed in Section 5.2.1.3, as applicable, based upon the degree of automation involved. It should be noted, however, that the lack of automation does not imply the lack of a need for configuration management of an E&V aid. In addition, Version Description Documents shall be prepared to identify new versions of each E&V aid.

5.2.2.4 Quality Control Requirements

Each E&V aid shall be developed in accordance with predetermined requirements. In addition, procedures shall be in place to appraise conformance to requirements, take corrective action when necessary, and plan for improvements in both the product and the process.

Each E&V aid shall be evaluated throughout its development in accordance with an appropriate methodology. The strength of the methodology chosen shall be dependent upon the APSE Evaluation and Validation Category supported by the aid. For example, aids which support the determination of conformance to a standard, or Category E formal validations, shall be evaluated in accordance with an extremely rigorous methodology. Aids which support subjective evaluations, or Category A evaluations, do not require the same degree of thoroughness during the evaluation process.

Each E&V aid shall undergo formal acceptance procedures that demonstrate the ability of the aid to satisfy all specified functional and quality requirements. When it is infeasible to formally demonstrate the satisfaction of requirements in this manner, sufficient evidence resulting from the appropriate analyses shall be available to indicate the aid's ability to satisfy requirements.

To ensure the suitability of the aid for the intended end-user, each E&V aid shall undergo beta testing conducted by a set of users that represent the spectrum of expected end-users. The purpose of this testing activity is to allow the determination of the suitability and effectiveness of the E&V aid in the operational environment.

Finally, each E&V aid shall undergo testing and evaluation by an independent group of individuals who are experts with respect to the objectives of, and the procedures, for the Evaluation and Validation of APSEs.

APPENDIX A

ACRONYMS

ACEC . . . . Ada Compiler Evaluation Capability

ACVC . . . . Ada Compiler Validation Capability

AJPO . . . . Ada Joint Program Office

APSE . . . . Ada Programming Support Environment

APSEWG . . . . APSE Analysts Working Group

AVO . . . . Ada Validation Organization

CAIS . . . . Common APSE Interface Set

CAISWG . . . . CAIS Working Group

CM . . . . Configuration Management

COORDWG . . . . . . Coordination Working Group

CVC . . . . CAIS Validation Capability

ECP . . . . Engineering Change Proposal

E&V . . . . Evaluation and Validation

GFE . . . . Government Furnished Equipment

KAPSE . . . . Kernel Ada Programming Support Environment

KIT . . . . KAPSE Interface Team

KITIA . . . . KAPSE Interface Team Industry/Academia

MAPSE . . . . Minimal Ada Programming Support Environment

REQWG . . . . Requirements Working Group

SEE . . . . Software Engineering Environment

SEVWG . . . . Standards Evaluation and Validation Working Group

RTS . . . . Run Time System

APPENDIX B

E&V TEAM REQUIREMENTS WORKING GROUP MEMBERSHIP

Chairperson :	Helen Romanowsky	Rockwell International
Vice-chair:	Marlene Hazle	MITRE Corp.
Members:	Jerry Brookshire	Texas Instruments
	Mike Burlakoff	University of Missouri
	Peter Clark	TASC
	Rich Fleming	Aerospace Corp.
	Robert Fritz	Computer Sciences Corp.
	Kathy Gilroy	Software Productivity Solutions
	Pat Lawlis	AF Institute of Technology
	Tim Lindquist	Virginia Tech
	Ronnie Martin	Georgia Tech
	Mike Meirink	Sperry Corporation
	John Miller	McClellan AFB
	Sandi Mulholland	General Dynamics
	Amos Rohrer	EG&G
	Ray Sandborgh	Sperry Corporation
	Nelson Weiderman	Software Engineering Institute

APPENDIX C

REFERENCES

1. Evaluation and Validation (E & V) Plan, Version 2.0. Air Force Wright Aeronautical Laboratories, Wright-Patterson AFB, 30 November 1984

2. "Taxonomy of Tool Features for a Life Cycle Software Engineering Environment."

3. The DACS Glossary, A Bibliography of Software Engineering Terms, October, 1979.

APPENDIX E

MINUTES

of the

EVALUATION AND VALIDATION (E & V) TEAM MEETING

4 - 6 December 1985

The task for the Evaluation & Validation of Ada* Programming Support Environments (APSEs) is sponsored by the Ada Joint Program Office (AJPO).

TABLE OF CONTENTS

1.0 WEDNESDAY, 4 DECEMBER 1985 . . . . E-3
- 1.1 Welcome, Introductions, and General Business. E-3
- 1.2 The Ada Run Time Environment Working Group (ARTEWG) Status Report E-3
- 1.3 Software Engineering Institute (SEI). E-5
- 1.4 E & V Classification Schema E-7
- 1.5 E & V Configuration Management Plan E-9

2.0 THURSDAY, 5 DECEMBER 1985 . . . . E-10

3.0 FRIDAY, 6 DECEMBER 1985 . . . . E-10
- 3.1 Introductions E-10
- 3.2 The Ada Joint Program Office (AJPO) E-11
- 3.3 Working Group Status Reports. E-14
  - 3.3.1 Coordination Working Group (COORDWG) Status Report E-14
  - 3.3.2 APSE Working Group (APSEWG) Status Report . . . E-14
  - 3.3.3 Standards Evaluation and Validation Working Group (SEVWG) Status Report E-15
  - 3.3.4 Requirements Working Group (REQWG) Status Report E-15
- 3.4 Software Engineering Institute, Revisited E-15
- 3.5 Announcements E-16
- 3.6 E & V Classification Schema, Revisited E-17
- 3.7 Closing Remarks E-17

APPENDIX A ACTION ITEMS AND RESOLUTIONS FROM THE SEPTEMBER E & V MEETING E-19

APPENDIX B ACTION ITEMS FROM THE DECEMBER E&V MEETING E-21

APPENDIX C ACRONYMS E-23

APPENDIX D LIST OF DOCUMENTS DISTRIBUTED AT THE DECEMBER E&V MEETING. E-25

APPENDIX E LIST OF ATTENDEES E-26

1.0 WEDNESDAY, 4 DECEMBER 1985

1.1 Welcome, Introductions, and General Business

The Evaluation and Validation (E & V) Team meeting began with opening remarks by chairperson Raymond Szymanski, followed by an introduction of the host, Marlow Henne, of the Harris Corporation.

Ray Szymanski welcomed the team and introduced the new people:

- LCDR Philip Myers from the Ada Joint Program Office (AJPO)

- Mr. Michael Mills, Aeronautical Systems Division (ASD/AXS) WPAFB, replacing Mr. Nelson Estes.

- Dr. Robert Loomis, Army Materiels Command

It was announced that:

- The Ada Run Time Environment Working Group (ARTEWG) has agreed to become a consulting arm to the E & V Team on Run Time Environment (RTE) issues.

- There are some problems with the Ada20 changeover. Those who do not have access to the MILNET can call Mr. Gil Austin at the Ada Information Clearinghouse.

- Beginning with this meeting, the E & V Team will have Army representation.

- Lt. James Kirkpatrick has relinquished his position with the Standards Evaluation and Validation Working Group (SEVWG), and Mr. Gary McKee has agreed to be acting chairperson this week.

The fourteen action items from the September meeting were reviewed. (The status of those items can be found in section 3.8.)

Speakers for Wednesday's session were announced: Kathleen Gilroy from the ARTEWG; Mr. Nelson Weiderman of the Software Engineering Institute (SEI); and Dr.Bard Crawford, Mr. Peter Clark, and Mr. Orville Branham of The Analytic Sciences Corporation (TASC).

1.2 The Ada Run Time Environment Working Group (ARTEWG) Status Report

Kathleen Gilroy
Software Productivity Solutions, Inc.

The purpose of the Ada Run Time Environment Working Group (ARTEWG) is to develop products and services for the Ada community. They are not an underwriter's lab; they are attempting to identify specific problems and to bring them to the attention of appropriate agencies. The ARTEWG is sponsored by SIGAda and endorsed by the AJPO. They are basically users

and implementors with the following objectives:

- Establish guidelines to promote reusability of Ada software

- Improve performance of Ada components

- Provide a means for evaluation/selection of RTEs

- Provide a mechanism for community interchange and interfacing

- Promote quality RTE implementations

- Identify/resolve Ada RTE issues

The ARTEWG approach toward attainment of these goals is divided into five basic tasks: 1) elaborate Ada implementation dependencies, 2) determine Ada implementation approaches, 3) define RTE requirements imposed by applications, 4) map requirements onto implementations, and 5) derive commonality of RTE interfaces.

An initial set of baseline products available to the public will include:

- A catalogue of implementation dependencies (currently in rough draft form)

- A catalogue of implementation approaches (no draft available)

- A specification of application requirements (transcripts from interviews in written form, but not available outside of the ARTEWG)

- Guidelines for use of Ada RTEs (not available)

- The catalogue of RTE interface options (some options will be available in July 1986)

Additional by-products include a file of Ada RTE issues, an "Ada RTE Transportability Handbook," Appendix F documentation requirements, a dictionary of RTE terminology, an annotated RTE bibliography, and an introduction to Ada RTEs. For documents as they become available on the NET, the ARTEWG-INFORMATION subdirectory can be accessed through the ADA-INFORMATION directory.

The ARTEWG, chaired by Mr. Mike Kamrad of Honeywell Systems and Research Center, consists of three major subgroups: 1) Implementation Dependencies headed by Darryl Winters of Sanders Associates, 2) Application Requirements headed by Dock Allen of Control Data Corporation, and 3) Common Interfaces headed by Charles McKay of the University of Houston-Clear Lake (UHCL). There are 20 to 30 principal members and a larger number of advisory members.

A simplified definition of a run time environment is that an RTE equals code conventions plus data structures plus predefined routines. The Ada compilation system brings in user libraries and a set of packages called the Run Time Library (RTL) and links them into the source code that is generated by the compiler. This is called the Run Time System (RTS).

The Ada compilation system is heavily dependent on the RTE and on how the RTE is defined. In APSE evaluations, the relationship between the functions and the RTE must be considered. When you have an APSE that is implemented in Ada, the RTE is going to impact the functionality, the performance, and the quality of the tools. The run time environment must be an integral part of APSE evaluation and validation.

The ARTEWG can benefit the E & V Team by developing products and tools that can be incorporated into E & V technology, by supplying documents for input to the E & V requirements, and by providing the manpower and expertise for heightening community awareness of E & V products.

The ARTEWG would like to see an information exchange with the E & V Team via status reports, the MILNET, and the Ada2O. There is the possibility of holding joint meetings to provide interaction in generating mutually beneficial technology.

1.3 Software Engineering Institute (SEI)

Mr. Nelson Weiderman
Software Engineering Institute

The Software Engineering Institute (SEI), located at Carnegie-Mellon University (CMU), is a federally funded research and development (R & D) center sponsored by the Department of Defense (DoD). They are a small, relatively new organization whose primary purpose is technology transition. The SEI will emphasize the following areas:

- Improving the quality of operational software

- Accelerating the process of getting the technology into practice

- Promulgating the use of modern techniques and methods

- Establishing standards of excellence in the software engineering practice

The SEI is one of three parts of the Software Initiative; the other two are the Software Technology for Adaptable Reliable Systems (STARS) and the AJPO. The SEI differs in that: 1) the SEI has a full-time, permanent staff, 2) all work is done inhouse--no subcontracting, and 3) the SEI does not compete with industry.

Basically, the SEI was established to identify the problems and needs of the software community in the DoD and industry, and to search the R & D labs of industry and universities for the technologies available to

solve these problems.

The SEI reports to both CMU and the DoD. They also interface with a board of visitors comprised of people from industry, government, and educational institutions. This board is appointed jointly by the director of the SEI, the president of CMU, and the AJPO.

The internal structure of the SEI is broken down into four organizational groups: research and education, technology exploration, technical and administrative services, and technology transition and training. One mechanism for the transition of technology is the SEI Affiliate Programs (industrial, government, and academic) which are just getting underway.

The SEI programs consist of a five-year plan addressing broad technical areas and a one-year plan listing a set of specific projects. These initial projects are underway in six basic areas:

1. Education. The major product here is the design, development, and dissemination of a Master of Science in Software Engineering (MSE) curriculum to increase the number of available software engineers. Projected deliverables include an MSE curriculum definition, a set of 30 to 40 one-credit modules issued as a monograph series (currently under negotiation with the publisher), and teacher training workshops, seminars, and symposia.

2. Technology Identification and Assessment. This project addresses the technology transition process. Monthly reports are being produced which cover distributed processing, user interface, tool interface, database, and environments.

3. Software Factory Workshop. This area deals with software development problems. Periodic workshops and one-day meetings are held for identifying, discussing, and reporting on approaches to software factories.

4. Showcase Environment. This project allows demonstration of various capabilities and environments. Current activities involve an infrastructure to support tool integration and a look at Interface Description Language (IDL) as possible support.

5. Evaluation of Ada Environments. Part of this project is the development of generic experiments focusing on what a user does independent of the tools he has available. These generic experiments will be translated into specific experiments for particular environments, producing both translation results and actual experiment results, which in turn will be the basis for an evaluation or analysis of those environments. An initial project is the production of some environment-independent products to be the evaluation technology.

6. Software Licensing. The emphasis here is on identifying DoD software acquisition and licensing problems. Some of the issues being examined are property laws, protection of software by copyright, the Data Rights Clause, and a recent law protecting semi-conductor chips. The lawyers undertaking this project have identified some of the problems, have viewed some case studies, and have submitted a report.

The Software Factory Workshop project, the Evaluation of Ada Environments project, and the Software Licensing project are of particular interest to the government; the remaining projects are internally generated. However, most of the information is applicable to everybody.

NOTE: This presentation is to be continued on Friday, 6 December 1985.

1.4 E & V Classification Schema

Dr. Bard Crawford
Mr. Peter Clark
The Analytic Sciences Corporation (TASC)

The Analytic Sciences Corporation (TASC) is the technical support contractor for the E & V Team, and is responsible for the production of a Classification Schema, a Reference Manual, and a Guidebook.

The Guidebook will contain the technology for implementation of tools and the techniques for evaluation and validation. This book will go out to the public and will be updated once a year for the next several years.

The Reference Manual will serve as a bridge between the user and the technology. A conceptual diagram presents the Reference Manual as an interactive system with various inputs and approaches which eventually lead to specific places in the Guidebook where the technology or technique is described.

The purpose of the E & V Classification Schema is to provide a framework and organization for the Reference Manual and to determine the design of the Guidebook. The proposed formal definition of the schema is as follows: The E & V Classification Schema is a multi-dimensional taxonomy, or set of axes, used to classify items that are the subject of evaluation and validation--that is, tools, tool sets, and APSEs to be assessed; and to classify the E & V process, particularly the steps and elements used to guide assessors to appropriate elements of E & V technology.

The first draft of the Classification Schema has been distributed and comments are requested.

The purpose of this presentation is to review the Classification Schema restating its rationale, and to preview the Reference Manual and the Guidebook. It is desirable to realize a common vision of the Reference Manual and to gain a consensus of opinion as to what should be put into the Guidebook.

In viewing the Reference Manual as an interactive system, primary elements will pair up with attributes. These element-attribute pairs will be used to identify specific criteria for assessment and to point to one of the five E & V categories, which in turn, point to a particular place in the Guidebook.

The term "primary element" is used to describe elements that, through analysis, serve the E & V purposes best. These are things that pair up with attributes and are accessible. Provision must be made to map a person's interest into one of these primary elements, so that the user can follow through the indexing schema and find the appropriate E & V technology. The Reference Manual is being developed primarily for the APSE user; however, there are views common to builders as well.

The two domains of E & V classification are referred to as the subject or APSE classification area and the process or E & V classification area. The elements describing the subject area are host and target environments, objects, functions, and implementation characteristics, while functions, implementation characteristics, attributes, and E & V categories define the process area. The APSE is viewed as a collection of tools and objects existing in and for various machine environments. In addition, tools are seen as being particular implementations of various functions.

In defining the schema axes, the focus is on function and implemention characteristics as separate issues. These are primary elements, whereas machine environments, objects, and tools are not. Host and target environments are not directly involved in the statement of E & V objectives; the tests are to evaluate the tools in the APSE. As for objects, it is difficult to establish standards which are relevant to the E & V process, and in the area of tools, the evolution is too rapid for E & V technology to keep pace.

The central schema creates an intersection of functions and implementations with attributes. One of the five classic E & V categories may be appropriate for each pairing of these functionally-dependent attributes or implementation- dependent attributes. One or more pairs from each group may be needed to find the entire set of E & V technology for any given tool or set of tools. A tool is defined by its function and implementation together.

The functional taxonomy has elements from both the National Bureau of Standards (NBS) and the Software Engineering Environment (SEE) taxonomies. It also contains some insertions from work done by Texas Instruments (TI). The attribute taxonomy contains some details from the REQWG's requirements document, and is made up of a functionally-related

part and an implementation-related part.

Outlines of the proposed Reference Manual and the Guidebook were presented to the team for review. For every entry in the Guidebook, whether it is a chapter or an entire volume, there will be a detailed, single-page summary in Appendix A, B, C, D, or E of the Reference Manual. These summary pages will point to the appropriate section of the Guidebook which will be organized into five major sections corresponding to the five E & V categories. Various chapters in the Reference Manual will have codes indicating which of the appendices contain the necessary summary pages.

NOTE: Team members are encouraged to submit entries for the Guidebook as soon as possible.

1.5 E & V Configuration Management Plan

Mr. Orville Branham
The Analytic Sciences Corporation (TASC)

Configuration Management (CM) is basically a discipline to apply technical and administrative direction and surveillance to development and tasking of documentation for systems that are being developed. This CM plan is based on MIL-STD-483, and has been tailored to the APSE E & V Team's charter.

Internal organization has the E & V Team functioning as a Configuration Control Board (CCB) comprised of the E & V Team chairperson, E & V Configuration Management Office (CMO) secretary, and the E & V Team working group chairpersons. The CCB will review documentation and ensure consistency with team views.

Configuration Management consists of four primary elements:

- Configuration Identification (CI). Responsibilities include establishing and maintaining a library, assigning CI numbers, and tracking documentation.

- Change Control. Responsibilities include administering control over changed documents, particularly baselined ones, and ensuring that future versions have been properly reviewed before release.

- Status Accounting. Responsibilities include keeping records and maintaining logs of documentation in process or in review.

- Technical Review and Audits. Primarily associated with programs that are under development.

Documents may be categorized as: 1) Internally generated for external communications (example: the Reference Manual), 2) Internally generated for team communication (example: the minutes), 3) Internally generated for description of team assets (example: the reference list),

and 4) Externally generated (example: documents produced by the STARS or the ARTEWG). These categories are proposed to aid the team in identifying and tracking documents. The functions and responsibilities of the CMO were noted on a flowchart presenting the approval flow of documents in each category.

A discussion followed on which documents would go into the library and under which category they would fall. Comments regarding the various levels of CM control and the availability of status information via the NET were accepted from the team.

In order to record and maintain the documentation generated by the E & V Team, a status tracking system is necessary. The only currently existing tracking system is on a VAX at WPAFB. Documentation approved by the team will be put into that system and be available for tracking. The system has a sorting capability whereby E & V documentation can be separately identified. There is the possibility of having a status report of such documentation accessible on the NET. This report will contain information such as which documents are currently being reviewed, expected dates for new drafts, etc. However, the first step is to collate the information for placement into the VAX database.

For the technical reviews and audits section of the CM plan, the procedures used are identified in the Avionics Division Configuration Management Plan with details found in MIL-STD-1521.

A question and answer period followed concerning what would fall under Configuration Management, how revisions to documents would be made, and the possibility of administrative support with a central site for making changes and generating new versions of a document.

NOTE: The draft CM plan was distributed to working group chairpersons, and comments are requested by 8 January 1986. A final draft plan incorporating those comments is expected by 31 January 1986.

The general session of the E & V Team meeting was adjourned. Individual working groups met for the remainder of the day.

2.0 THURSDAY, 5 DECEMBER 1985

The entire day was devoted to individual working groups. Excerpts from the various working groups may be available sometime in the future.

3.0 FRIDAY, 6 DECEMBER 1985

3.1 Introductions

Chairperson Raymond Szymanski reopened the general session. Marlow Henne introduced Miriam Martinez, section chief of the technology section of the Harris Corporation. After some general comments, Raymond Szymanski then introduced the speaker for the morning, Virginia Castor, Director of the Ada Joint Program Office (AJPO).

3.2 The Ada Joint Program Office (AJPO)

Virginia Castor
Director of the AJPO

[Portions of this presentation were given at the November '85 SIGAda meeting in Boston.]

The object of this presentation is to increase the team's awareness of the many facets of the Ada Program by giving an overall view of AJPO activities and involvements. The E & V Team is getting a significant amount of recognition, both within the United States and in the international community. The team is being looked to for the technology associated with environments, and the AJPO will do what it can to support and promote this effort. The AJPO sponsor for the E & V Team is LCDR Philip A. Myers. The main task areas of the AJPO are standardization of the language (control and validation) and use of the language (education, training, promotion, etc.).

Ada is an ANSI/MIL-STD language, and became a Federal Information Processing (FIP) standard in October of this year. Canada, Germany, the United Kingdom (UK), and Sweden are adopting Ada, and NATO has selected Ada as a single high order language for those participating nations that come under Command Control Communications Intelligence (CCCI).

The Ada Joint Program Office has been working closely with the International Organization for Standardization (ISO) to have Ada adopted as an international standard. The international community is especially interested in the Ada Compiler Evaluation Capability (ACEC). This is an area of leverage for promoting the E & V task.

The Ada Compiler Validation Capability (ACVC) is approximately 2500 tests. There are validation facilities at the WPAFB, the GSA in Washington, France, Germany, and the UK. The ACVC is under configuration control at the WPAFB. There are a number of validated compilers, both commercial and military, and the validation policy is currently under revision.

In the area of education and training, another Tri-Service team has been established: the Ada Software Engineering Education and Training (ASEET) Team. The Armed Forces Communications and Electronics Association (AFCEA) initiated a requirement study to determine training needs in Ada for industry. Information is available in the form of a Catalog of Resources in Education for Ada Software Engineering (CREASE) and a video tape on the Ada programming language distributed by the U.S. Army.

The Ada Information Clearinghouse (AdaIC) provides general information and promotional material for Ada via an online Ada-INFORMATION directory, telephone queries, and information mailings. Everyone should get on the distribution list and receive the AdaIC newsletter in order to keep abreast of Ada events.

It was noted that the current contract for the Ada Information Clearinghouse terminates shortly and there will be a competitive procurement for a support contractor.

In the area of Ada technology insertion, math library work is being conducted to develop an analogous set of math routines for Ada. The SIMTEL-20 repository is now in White Sands, New Mexico, but it can be accessed through Ada-INFORMATION. The AJPO is attempting to consolidate information and to make it more readily available. One large major repository subdivided into accessible categories is one goal of the AJPO.

The AJPO is currently collecting information on every program and every system that is using Ada. They are soliciting input not only from the DoD but from industry as well. Anyone willing to share this type of information is encouraged to contact the AJPO. Current facts and figures are important elements in promoting the use of Ada.

There has been a lot of activity within the AJPO. Because of the work load and a new staff, responsibilities in the task areas have shifted somewhat.

- LTC Taylor is the Army deputy, and he is responsible for all Ada education and training activities. He is also the international representative from the AJPO.

- LCDR Myers is handling the area of Ada environments and will be coordinating all activities and issues in this field.

- MAJ Kopp is in charge of Ada promotion. He will be gathering and compiling information to be placed in a central repository.

- Paul Cohen is involved in Ada validation--formal semantics, verification issues, etc.

- Burt Newlin is active in the standards arena, primarily dealing with the CAIS.

The Ada Board was established in 1983 under the Federal Advisory Committee Act as an advisory board to the AJPO. Formal approval from persons in the White House is necessary for membership. This approval is presently being processed, and if possible, the first official Ada Board meeting will be convened in February at the SIGAda meeting. At this point in time, there is no official membership to the Ada Board, but Raymond Szymanski, chairperson of the E & V Team, and Trisha Oberndorf, chairperson of the KIT/KITIA, are included in the nominations.

The DoD validation policy on Ada maintains that there must be formal standardization of the language--no subsets, supersets, or dialects. "Ada" is a registered trademark of the U.S. Government, and any task sponsored by the AJPO must indicate this. Use of the trademark means

compliance with the standard. Within the validation process, the AJPO oversees the language and ensures conformance to policy.

A letter from the Under-Secretary of Defense Research and Engineering affirmed the mandatory use of Ada in all mission critical systems effective July 1984. The DoD directive 5000.29, currently under revision, will replace the draft DoD 5000.21 and will incorporate Ada policy.

Problems concerning validation issues include criticisms on: the lack of a formal validation policy with regard to Ada compilers; the expense and complexity of the validation process; the policy on the use of validated compilers on DoD programs and Ada compilers for restricted targets; the inconsistency in validation summary reports; and what information should be on a validation certificate.

In order to alleviate some of these problems, the AJPO is approaching Ada validation from three areas: 1) policy on validation of Ada compilers, 2) procedures on policy implementation, and 3) guidelines for DoD programs (consistent with the DoD directive 5000.29). Separate documents for each area have been drafted and are being reviewed by AJPO staff members.

The validation policy covers validation of base compilers, registration of derived compilers, and maintenance of derived compilers. The validation of a compiler will consist of the validation of a base compiler and a base configuration. The full ACVC will be applied. A validation summary will accompany this, and a validation certificate will be issued. A list of the validated compilers will be maintained by the AJPO.

A list of registered derived compilers will also be maintained by the AJPO. A registered derived compiler is derived from a compiler that has received full validation. A vendor may request registration of a compiler that has basically the same kind of configuration as a validated one. By doing this, the vendor is affirming that the compiler will run in a particular system and that it conforms to the standard.

This claim can be challenged at any time. If the claim is proven false, the derived compiler is not taken off the list, but the successful challenge is noted. Consequently, the compiler is no longer validated, and the vendor's credibility could be in jeopardy in the entire Ada community.

The AJPO would like a courtesy copy of any challenge, but the actual procedures have not been finalized. The initial interchange is between the challenger and the vendor who has made the claim, and nothing will be publicized until it has been successfully shown that the derived compiler has failed one of the ACVC tests that the original system executed successfully.

As far as the maintenance of these compilers is concerned, if the vendor states that the maintenance has not affected the performance, it is still considered a validated compiler. This has always been in effect. The total validation policy will be properly reviewed before being released.

This presentation was brought to a close with the reading of a letter from Donald A. Hicks, Under-Secretary of Defense Research and Engineering, regarding the implementation of Ada in DoD programs. Copies of this memorandum were made available for team distribution.

3.3 Working Group Status Reports

3.3.1 Coordination Working Group (COORDWG) Status Report

The COORDWG chairperson, Don Jennings, reported no changes in personnel. Deliverables due this quarter were the E & V Status Report and the September minutes. Accomplishments include the E & V Status Report which is being reviewed and will go on the NET in late December, a review of the September minutes, and a review of the Configuration Management Plan with proposed changes to the document approval flowchart. The key issue addressed this quarter was the need for the E & V Team's coordination with the ARTEWG. No unresolved problems or action items were noted. Projected work for next quarter consists of producing the E & V Status Report and the December minutes. Review may begin on Version 3.0 of the Public Coordination Strategy Document which is due the following quarter. There are no deliverables due next quarter, and no presentations are planned for the next meeting. Reminder: anyone giving a briefing concerning E & V is required to fill out a template.

3.3.2 APSE Working Group (APSEWG) Status Report

The APSEWG chairperson, Elizabeth Kean, reported there had been no change in personnel, but Guy Taylor will not be attending future E & V meetings. Version 2.0 of the APSE Analysis Document is the deliverable due this quarter. The document will be in Elizabeth's Ada20 directory next week and ready for final team review. One of the major changes in this version is the incorporation of the SEE taxonomy. THIS DOCUMENT IS NOT TO BE DISCLOSED OUTSIDE OF THE E & V TEAM. Ten working days will be allotted for review. The key issues addressed this quarter were the dissemination restriction on the Analysis Document and a proposed survey of APSE functions. The only unresolved problem is the restriction on the document, which will be resolved by Philip Myers. Projected work for next quarter: 1) finalizing a format for a survey of APSE functions, 2) running some IDA benchmark tests on the ALS for validity and possible fusion into E & V technology, and 3) adding the REQWG attributes to the next version of the APSE Analysis Document. There are no deliverables due next quarter and no presentations planned for the next meeting. Action items for individual group members were noted.

3.3.3 Standards Evaluation and Validation Working Group (SEVWG) Status Report

The SEVWG report was given by acting chairperson Gary McKee, who announced the addition of Michael Mills to the group. There were no deliverables due this quarter. Accomplishments include a draft of a Components Validation Procedures (CVP) Document which is ready for Ray Szymanski's review, an early draft of a CAIS Analysis Document (CAD), an interview with Kathleen Gilroy of the ARTEWG, and a meeting with Virginia Castor, Director of the AJPO. Key issues addressed this quarter were the CVP, the CAD, the CAIS, and the Classification Schema. Projected work for next quarter is a draft of the CAIS Analysis Document.

3.3.4 Requirements Working Group (REQWG) Status Report

The REQWG chairperson, Patricia Lawlis, announced no personnel changes. Version 2.0 of the Requirements Document was the deliverable due this quarter, and it is now ready for team review. There is also a draft of the Tools and Aids Requirements Document ready for comments. (NET versions of these documents are expected by 17 January.) Other accomplishments include coordination with the ARTEWG on runtime evaluation issues, further definition of whole APSE issues as opposed to individual components, and preliminary results of a survey to assess priorities for evaluator development. Several key issues were addressed. The team's relationship with contractual efforts such as the Ada Compiler Benchmark test suite and the ACEC was discussed. There was an initial look at the Classification Schema. The need for additional funded support was considered, especially in the areas of word processing for updating documents and a mechanism for quick production of Strawman documents. The major unresolved problem is getting the documents and reports on the NET. During the next quarter, the group will be taking a closer look at the availability and current efforts of E & V technology and various methods of evaluating that technology. There are no deliverables due next quarter and no presentations planned for the next meeting. The group is recommending: 1) a clarification of how to address team mail on the NET, 2) a presentation by Jon Squire on the status of the Performance Issues Working Group (PIWG), and 3) a collection of quarterly NET mail made available in hard copy for distribution at team meetings.

3.4 Software Engineering Institute, Revisited

Mr. Nelson Weiderman, SEI

Nelson Weiderman presented some of the philosophy and scope of the APSE evaluation project at the SEI.

The basic idea is to limit the scope of the evaluation, to work on particular subsets of a problem, and to come up with something systematic and reproducible. The objective is to develop an extensible, experimentally based test suite to apply to different APSEs. The SEI

advocates rapid prototyping for evaluation--refining various parts as they progress. It is desirable to build on previous evaluation work done by various other organizations.

In order to narrow the scope of APSE evaluation, the focus is on host rather than target issues, on the process of developing software as opposed to what happens after development. Environmental issues are considered over compiler issues, primarily to avoid redoing work that has been done, for example, in the ACVC and the IDA benchmarks. Little time will be spent on correctness and validation issues. Ideas on functionality will be based on exemplary systems. Subjective judgment comes into play, but the APSEs can be compared to those environments which have been sufficiently experienced. The main focus will be on primary tools and features.

Several systems are currently under consideration for evaluation including the ALS; Verdix Ada and UNIX; DEC Ada and VMS; and the Rational. The four principal areas of investigation are functionality, performance, user interface, and compatibility with the underlying system.

3.5 Announcements

- There will be an investigation to determine reasons for the lack of bidders on the CAIS Validation Capability (CVC). The RFP will be re-released in late January 1986.

- An April CBD announcement is planned for an RFP with respect to the ACEC effort.

- Arrangements will be made for a Birds of a Feather session at the February meeting of the SIGAda.

- The March E & V Team meeting is presently scheduled for Dayton, Ohio; however, a West Coast site is under consideration. Ray Szymanski will make the final decision.

- If Virginia Castor attends the Ada Europe meeting in January, she will brief the E & V Team as to common areas of interest and the feasibility of E & V Team involvement.

- The AJPO will be signing out letters of appreciation endorsed by Raymond Szymanski to be sent to the various organizations of the E & V Team members.

- The contractual capacity of the Systran Corporation will be reviewed for possible additional administrative support for working group documents.

- Version 2.0 of the APSE Analysis Document has incorporated the SEE taxonomy, thus making it subject to the same control and restriction, i.e., International Traffic in Arms Regulation (ITAR).

- Regrets were expressed over Guy Taylor and John Miller leaving the E & V Team.

- A new contractual administrative support person, Barbara Rhoads from Systran Corporation, was welcomed to the group.

- Handouts from this meeting will be mailed soon to all attendees.

- The March meeting might consist of three full-day sessions.

3.6 E & V Classification Schema, Revisited

Mr. Bard Crawford (TASC)

TASC will be analyzing the substantial amount of Classification Schema feedback received this week from meetings with the APSEWG and the REQWG, and from individual conversations.

This additional time provided another look at the two divisions of primary elements (function and implementation characteristics) and how they are used to identify the appropriate E & V category. Speed of compilation was cited as an example of a functionally-independent attribute: the attribute is speed, the function is compilation, and evaluation of that combination falls under E & V category B (no standard exists, but tests are available for an objective evaluation). Issues will be addressed as they come up to see into which primary element category they fit.

Some uncertainty has been expressed over the terminology used in the schema; however, the basic question is whether or not this process will indeed provide an adequate guide into E & V technology. Numerous examples are needed to ensure against an important issue not being covered by either of the two domains. Both the Reference Manual and the Guidebook have an open-ended design to allow for growth.

Specific examples are being solicited for write up in the Guidebook. Current candidates for inclusion are ARTEWG issues/guidelines, editor evaluation, IDA benchmarks, ACEC, CAIS Operational Definition, ARTEWG standards, ACVC, and CVC. It is desirable to produce some version of these documents as soon as possible.

NOTE: TASC would like any additional comments on the Classification Schema by late December. The schema will be reissued one more time (in draft form).

3.7 Closing Remarks

- Various team members extended congratulations to Virginia Castor for her work as the AJPO Director.

- Virginia Castor announced a December briefing she is giving before the Investigatory Subcommittee of the House Appropriations Committee.

- The AJPO is working toward Congressional support for the Ada Program.

- The AJPO in conjunction with the SEI is generating an aggressive educational thrust toward enlightening more persons as to what Ada is and the advantages Ada has to offer.

The E & V Team meeting for December 1985 was adjourned.

APPENDIX A

ACTION ITEMS AND RESOLUTIONS FROM THE SEPTEMBER E & V MEETING

	ITEM	STATUS
AI-9-6-85-1	Systran. Compile a list of all documentation distributed at the September meeting and include it in the minutes.	Accomplished
AI-9-6-85-2	Systran. Implement CM on the documents distributed at the meeting.	Accomplished
AI-9-6-85-3	Szymanski. Archive the Team mail.	Pending
AI-9-6-85-4	Szymanski. Locate and make available E&V Team viewgraphs.	Carried over
AI-9-6-85-5	Szymanski. Open NET accounts for Nelson Weiderman, SEI and Peter Clark, TASC. Change P. Dobbs account to S.L. Mulholland, and Bud Hammond's account to Jeff Facemire.	Accomplished as part of mass changeover
AI-9-6-85-6	Szymanski. Investigate meeting with Ada Europe.	Resolution due by 27 January
AI-9-6-85-7	Szymanski. Investigate the legality of the survey proposed by REQWG on commercial environments.	Under investigation
AI-9-6-85-8	Szymanski. Arrange for the STARS Methodology Team to give a presentation at the December meeting.	Pending for future meeting
AI-9-6-85-9	Szymanski. Consult with ITARS and J. Castor on the Public Review problem.	No comment made
AI-9-6-85-10	Szymanski. Consult with STARS to see if methodology should be included in the E & V charter.	Pending

AI-9-6-85-11	Harto. Send a message on the NET telling where to send visit requests for the December meeting.	Accomplished
AI-9-6-85-12	Jennings. Send the E & V Status Report to R. Szymanski at the KIT/KITIA meeting.	Accomplished
AI-9-6-85-13	Fritz. Put the Draft Tools and Aids Requirements Document on the NET.	Pending
AI-9-6-85-14	Fleming, Lawlis. Put the Draft Requirements Document, Version 2.0 on the NET.	Pending

APPENDIX B

ACTION ITEMS
FROM THE
DECEMBER E & V MEETING

AI-12-6-85-1	Systran. Mail presentation material to attendees.
AI-12-6-85-2	Systran. Put draft of December minutes on the NET.
AI-12-6-85-3	Systran. Prepare minutes of the REQWG and the APSEWG meetings.
AI-12-6-85-4	Szymanski. Schedule a meeting between the ARTEWG and the E & V Team for SIGAda.
AI-12-6-85-5	Szymanski. Establish contact with Space Command, Colorado Springs, Colorado.
AI-12-6-85-6	Szymanski. Send thank you note to Harris Corp. for hosting.
AI-12-6-85-7	Szymanski. Give briefing on E & V at SIGAda.
AI-12-6-85-8	Szymanski. Arrange for E & V Birds of a Feather meeting at SIGAda.
AI-12-6-85-9	Szymanski. Track Ada Europe interest in E & V effort. Investigate E & V participation in May Ada Europe Conference.
AI-12-6-85-10	Szymanski. Investigate support contractor capacity for additional administrative support for working groups.
AI-12-6-85-11	Szymanski. Find out what the "National Test Bed" is.
AI-12-6-85-12	Szymanski. Develop company reaffirmation letter for REQWG members or any team member if requested.
AI-12-6-85-13	Szymanski. Provide a detailed agenda for next meeting to include a list of deliverables and the status of each working group in relation to the schedule.

AI-12-6-85-14 Szymanski.
Seek additional active members for the SEVWG.

AI-12-6-85-15 Szymanski.
Investigate methods of putting documents onto diskettes for Systran to format and print.

AI-12-6-85-16 Szymanski.
Review the APSE Component Validation Procedures Document and send comments to John Reddan.

APPENDIX C

ACRONYMS

ACEC	Ada Compiler Evaluation Capability
ACVC	Ada Compiler Validation Capability
AdaIC	Ada Information Clearinghouse
AJPO	Ada Joint Program Office
APSE	Ada Programming Support Environments
APSEWG	APSE Working Group
ARTEWG	Ada Run Time Environment Working Group
CAIS	Common APSE Interface Set
CCB	Configuration Control Board
CM	Configuration Management
CMU	Carnegie-Mellon University
COORDWG	Coordination Working Group
CVC	CAIS Validation Capability
DoD	Department of Defense
E&V	Evaluation and Validation
IDA	Institute for Defense Analyses
IDL	Interface Description Language
ITAR	International Traffic in Arms Regulation
KIT/KITIA	KAPSE Interface Team/from Industry and Academia
NBS	National Bureau of Standards
PIWG	Performance Issues Working Group
REQWG	Requirements Working Group
RTE	Run Time Environment
SEE	Software Engineering Environment

SEI . . . . Software Engineering Institute

SEVWG . . . . Standards Evaluation and Validation Working Group

SIGAda . . . . . . Special Interest Group Ada

STARS . . . . Software Technology for Adaptable Reliable Systems

TASC . . . . The Analytic Sciences Corporation.

APPENDIX D

LIST OF DOCUMENTS DISTRIBUTED AT THE DECEMBER E & V TEAM MEETING

1. Agenda for 4-6 December 1985 team meeting
2. Materials used in "E & V and the ARTEWG"
3. Presentation materials used in "Software Engineering Institute: Overview"
4. Presentation materials used in "E & V Classification Schema: Report to E & V Team Meeting"
5. Presentation materials used in "Configuration Management for Ada Programming Support Environment Evaluation and Validation Team"
6. Presentation materials used in "The Ada Program"
7. 2 December 1985 memorandum from Donald A. Hicks
8. Attendance list
9. E & V Status Report for September 1985
10. Minutes of September 1985 E & V Team meeting

The E & V Classification Schema Report Draft Version 1.0 was made available to the team members prior to the meeting.

APPENDIX E

LIST OF ATTENDEES

Branham, Orville
TASC
3040 Presidential Drive
Fairborn, OH 45324

Brookshire, Jerry R.
Texas Instruments
MS 3114, P.O. Box 660246
Dallas, TX 75266

Castor, Virginia
Ada Joint Program Office
Room 30139 (Fern Street/C107)
The Pentagon
Washington, DC 20301-3081

Clark, Peter
TASC
1 Jacob Way
Reading, MA 01867

Crawford, Bard
TASC
1 Jacob Way
Reading, MA 01867

Deese, Capt. Albert J.
ASD/SIOL
WPAFB, OH 45433

Fleming, Richard
The Aerospace Corp.
M1/112, P.O. Box 92957
Los Angeles, CA 90009

Fritz, Robert
Computer Sciences Corp.
4045 Hancock St.
San Diego, CA 92110

Gicca, Gregory A.
GTE/WIS
1 Federal St.
Billerica, MA 01821

Gilroy, Kathleen
SPS, Inc.
P.O. Box 361697
Melbourne, FL 32936

Harto, Debra L.
AFATL/DLCM
Eglin AFB, FL 32542

Hazle, Marlene
Mitre Corp.
Burlington Rd.
Bedford, MA 01730

Henne, Marlow
Harris Corp.
505 John Rhodes Blvd.
Bldg. 1, R1/1167
Melbourne, FL 32901

Jennings, Donald
OC-ALC/MMECO
Tinker AFB, OK 73145-5990

Kean, Elizabeth
RADC/COEE
Griffiss AFB, NY 13441-5700

Lawlis, Patricia
AFIT/ENC, Bldg. 640
WPAFB, OH 45433

Lindquist, Timothy E.
Arizona State University
Computer Science Dept.
Tempe, AZ 85202

Loomis, Robert
HQ AMC
AMCOE-SB-S
5001 Eisenhower Avenue
Alexandria, VA 22333

Maher, Patrick
Magnavox
TC-10-C3 D542
1010 Production Rd.
Fort Wayne, IN 46808

Martin, Ronnie
Georgia Institute of Technology
Software Engineering Research Center
Atlanta, GA 30324

McKee, Gary
Martin Marietta Aerospace
M/S 0423, P.O. Box 179
Denver, CO 80201

Miller, John
SM-ALC/MMEHD
McClellan AFB, CA 95652-5609

Mills, Michael
ASD-AFALC/AXTS
WPAFB, OH 45433

Mulholland, S. L.
General Dynamics
6100 Western Pl., Suite 735
Ft. Worth, TX 76107

Myers, Philip A.
Ada Joint Program Office
The Pentagon
Washington, DC 20363

Reddan, John
Syscon Corp.
3990 Sherman St.
San Diego, CA 92110

Rhoads, Barbara L.
Systran Corp.
4126 Linden Avenue
Dayton, OH 45432

Romanowsky, Helen
Rockwell International
400 Collins Road NE, 124-211
Cedar Rapids, IA 52498

Shirley, Jane
Systran Corp.
4126 Linden Avenue
Dayton, OH 45432

Szymanski, Raymond
AFWAL/AAAF-2
WPAFB, OH 45433

Taylor, Guy
FCDSSA
Dam Neck
Virginia Beach, VA 23461

Weiderman, Nelson
SEI
Carnegie-Mellon University
Pittsburgh, PA 15213

Williamson, James
AFWAL/AAAF-2
WPAFB, OH 45433

Wills, Betty
AFCC/CCSO
Tinker AFB, OK 73145

APPENDIX F

MINUTES

of the

EVALUATION AND VALIDATION (E&V) TEAM MEETING

5 - 7 MARCH 1986

The task for the Evaluation & Validation of Ada* Programming Support Environments (APSEs) is sponsored by the Ada Joint Program Office (AJPO).

*Ada is a Registered Trademark of the U.S. Government (Ada Joint Program Office)

TABLE OF CONTENTS

SECTION PAGE

1.0 WEDNESDAY, 5 MARCH 1986 . . . . F-3
1.1 Welcome, Introductions, and General Business F-3
1.2 Human Factors Engineering (HFE). F-3
1.3 Portable Common Tool Environment (PCTE). F-6
1.4 E&V Reference Manual and E&V Guidebook F-8

2.0 THURSDAY, 6 MARCH 1986. . . . . F-11
2.1 APSEWG Survey. F-11

3.0 FRIDAY, 7 MARCH 1986. . . . . F-11
3.1 Remarks. F-11
3.2 Procurement Status F-11
3.3 General Business F-12
3.4 TASC Update. F-13
3.5 Working Group Status Reports F-13
3.5.1 Coordination Working Group (COORDWG) Status Report F-13
3.5.2 APSE Working Group (APSEWG) Status Report F-14
3.5.3 Requirements Working Group (REQWG) Status Report F-14
3.5.4 Standards Evaluation and Validation Working Group (SEVWG) Status Report F-15
3.6 Closing F-15

APPENDIX A Action Items and Resolutions from the September E&V Meeting. F-16

APPENDIX B March Action Item List F-18

APPENDIX C Acronyms F-19

APPENDIX D List of Documents Distributed at the December E&V Meeting F-21

APPENDIX E List of Attendees. F-22

1.0 WEDNESDAY, 5 MARCH 1986

1.1 Welcome, Introductions, and General Business

The Evaluation and Validation (E&V) Team meeting began with opening remarks by chairperson Raymond Szymanski, followed by the introductions of new people:

- Mary Tompkins, Lockheed Missiles & Space Company, Austin Division

- Dorothy John, AFWAL/AAAF-1, WPAFB

- Matt Emerson, Naval Avionics Center (NAC), Indianapolis, Indiana

Mary Tompkins is replacing Manda Sury and will be giving a presentation at the June E&V meeting to qualify for team membership.

It was announced that:

- Gary McKee is now chairperson of the Standards Evaluation and Validation Working Group (SEVWG), and the vice-chairperson is Mike Mills.

- Helen Romanowski will be taking over as the Requirements Working Group (REQWG) chairperson when Pat Lawlis leaves this summer.

- Betty Wills of the Coordination Working Group (COORDWG) will be making the viewgraphs for the team presentations at the Ada Europe Conference in May.

- Systran Corporation may provide some additional administrative support, if needed, as part of an AFWAL support contract.

Ray Szymanski then introduced the speakers for Wednesday's session: Mr. Robert Richards from EG&G Idaho, and Mr. Herm Fischer from Mark V Business Systems.

1.2 Human Factors Engineering (HFE)

Mr. Robert Richards
EG&G Idaho, Inc.

Bob Richard's presentation was titled "The Relevance of Human Factors Engineering to the Design and Evaluation of Ada Programming Support Environments," and the key topics for discussion included: human factors engineering (HFE); application of HFE to software development; and specific application of HFE to Ada Programming Support Environments (APSEs). These topics covered some of the basics of HFE, particularly as applied to software engineering and as related to Ada interests.

A primary HFE concern is that as systems become more complex, the cabilities may be reached. The APA Monitor official newsletter for the American Psychological Society states:

"We seem to be very close to, if not already beyond, the practical limitations of the human senses."

The Human Factors Society, made up of various technical groups, has been in existence for 29 years. This society came about as a result of the massive development of new machines that needed human operators, but today the emphasis is on man-computer interfacing. At a recent conference there were discussions relevant to software engineering and computer systems in areas such as speech interaction, prototyping, and expert/knowledge-based systems.

In an effort to define human factors engineering, a past president of the Human Factors Society indicated that HFE applies and creates--or creates and applies--information about human ability, limitations, and other characteristics to the design of systems, machines, environments, tools, jobs, and tasks for safe, effective, and comfortable human use.

In applying HFE to APSEs, the same basic approach can be used as in any software development effort. The earlier that HFE can be applied, the more productive and effective the system, from analysis through maintenance. This is an exciting and important research and development area for human factors engineering.

HFE applies to software development in various ways. In a software-based system, there is always the danger of human error. HFE facilitates performance that is less error prone. Many times this prevention can be built into the system, although automation does not necessarily reduce the human work load. Even though functions are being increasingly automated, there are still certain problems that humans can solve better and more efficiently than machines. However, machines should do those parts which they can do best to simplify the work of the operators.

Users' acceptance and satisfaction is another important area of HFE application. By looking at the human operator's processing, procedures can be streamlined initially and the system can be made more efficient from the start.

In applying HFE to systems development, it is necessary to look at a system specification in terms of the human implications (the training, personnel, and interface required) to avoid creating simply a hardware- or software-driven system. There are systems that achieve the assigned functionality, but are not tuned to an operator's needs.

Many people think that human factors engineering is not needed until after the concepts have been formulated, but actually the largest impact can be achieved at the start. In the requirements phase, functions/tasks are allocated to the computer, the operator, or both.

The human tasks and procedures are described, and the operator's information requirements are listed. It is important in this phase to develop a system concept that is consistent with an operator's view of the system's purpose.

There are many factors to consider during the design phase, such as the experience and skills of the operator; the use of menus versus commands; advantages and disadvantages of various dialogue types; the type of transactions; and the frequency of use. HFE can be a factor in determining target operators and their skill level in order to effect specific interface design. Hardware options are also considered during the design phase, and optimal devices are selected according to customer and environmental constraints.

Questions have arisen in the area of standards and in the application of guidelines or rules. A list of guidelines has been compiled, and it has been suggested that these guidelines be used as a baseline for the incorporation of specific rules. Experience has shown that guidelines cannot be enforced whereas specific rules can be. Establishing such rules is a possible future project.

Another stage of development is the prototype phase. Prototyping helps debug both the interface and the basic functions. HFE provides software personnel with a better conceptualization of the system. Potential users provide feedback in areas such as ease of operation, understandability, and consistency. Prototyping facilitates training and provides a customer with something concrete to see and touch.

During development, human factors engineers monitor the consistency of systems, in addition to providing continued consulting support in various areas. Some areas of HFE application within the system testing phase are error identification, recommendation of needed changes, and solicitation of user responses.

In the maintenance phase, many of the problems have human implications, but unless a major overhaul is needed, there is not much significant work that can be done by human factors engineers. At this stage, the product is improved or upgraded by job aids rather than by actually working with the interface. A good slogan for an effective HFE time frame is: the earlier the better. Those systems that detail and include the operator from the beginning tend to be the most economical.

Expert systems, documentation, and training are some other relevant areas of interest in the HFE field. Training can be a completely separate area, but much of the information needed by human factors engineers is also needed by designers in the development of basic instructions and any kind of embedded training.

In applying HFE to APSEs, one approach is to evaluate existing APSEs by developing some tailored evaluation instruments such as questionnaires and observation forms. Even though utilizing the evaluation instruments by checklisting, applying guidelines or rules, and observing task

performance is a valid activity, the amount of impact is minimal compared to the cost at this stage. Ideally, it would be more useful to incorporate HFE in the concept development of a new APSE.

This presentation was concluded with some views on applying HFE to new APSE development, specifically in the areas of editors, debuggers, naming conventions, command languages, and user-specific tools. In new APSE development, the user interface should be as consistent as possible across the components.

The following points were brought out during a question and answer period that followed:

- There is a need for human factors engineers to get into the system development process itself. This would allow a greater degree of interaction with software engineers and their subject matter.

- In addressing documentation strategies, there are no concrete answers. For example, some people may do better with a visual approach and others with a linear structure approach.

- There has been an attempt to insert an information system design phase into the life cycle which would precede the software/hardware phase.

- A formal HFE plan exists that specifies all the various stages and is submitted with every contract.

- To elicit the necessary resources, the subject of HFE needs to be considered an integral part of system development, rather than a sideline.

1.3 Portable Common Tool Environment (PCTE)

Mr. Herm Fischer
Mark V Business Systems

Herm Fischer began his presentation with some facts about the Portable Common Tool Environment (PCTE). PCTE is a set of C interfaces for UNIX that adds functionality similar to the Common APSE Interface Set (CAIS). The PCTE is a prototype implementation developed by a European consortium.

The French implementation of the PCTE is termed Emeraude. This project is partially funded by the French government in a joint venture with the GIE, a three-company partnership consisting of Bull, Syseca, and Eurosoft. The Emeraude schedule is conducted parallel to the PCTE and shares some of the same coding.

Ideas for the PCTE came primarily from a study done at Bull (ALPAGE) and one done by a United Kingdom Consortium (M-CHAPSE). Other influences include the CAIS, UNIX, and the Portable Ada Programming System (PAPS) design. The PAPS project is an Olivetti project and may become the basis of a portable piggyback in Ada.

Portability was the main concern of the PAPS, where as the PCTE project is concerned with compatibility with existing tools and is strongly tied to UNIX. The PCTE program is multilingual versus Ada only.

There are six companies involved in the PCTE project, the consortium in joint venture with the European Esprit program: Bull (France), ICL (UK), Siemens (Germany), Olivetti (Italy), GEC (UK), and Nixdorf (Germany). These companies match their funds to consortium funding, which results in certain ownership rights. These rights may be relinquished if a partner does not utilize resulting products within a 2-year period.

There are some commonalities within the three PCTE implementations (the Esprit program, the Emeraude project,and the Ada piggyback by Olivetti), and it is possible for the same person to be involved in two different projects.

The PCTE is working toward a completely homogeneous system built into the architecture of the kernel--a set of host workstations on a local area network (LAN) with shared resources. These may be different pieces of hardware running different UNIX kernels. The logical structure of a PCTE kernel was depicted as UNIX plus basic mechanisms plus a CAIS-like object management system. There is a sensitivity toward preserving all the utilities with UNIX so that the system remains useful as more advanced tools are being developed.

A functional division of the logical components of an interface set is similar to that of a kernel and consists of basic mechanisms, an object management system, distribution, and user interface.

One interesting part of user interface is windowing. Both Carnegie-Mellon University (CMU) and the Massachusetts Institute of Technology (MIT) are working on projects to develop windowing systems to be added on the UNIX.

At this point, Herm asked for any questions or comments from the audience. The issue of whether to standardize on the CAIS or on the UNIX was brought up, which led to a discussion on the types of problems that each is trying to solve. Tool portability is a major concern.

When comparing the PCTE and the CAIS, both similarities and differences can be seen. Similarities are in areas such as node models, relationship models, attributes, installation of process models, and Ada I/O. The PCTE differs from the CAIS in areas of schemas, distributed processing, compatibility with existing tools, user interface, and security. Some PCTE shortcomings are due to limitations in attributes

handling, intent codes, and querying. There are also some lower level concerns in kernel size, word size, and user interface.

The subject of PCTE can cause confusion because of the many countries involved, and the number of companies participating in parallel areas of the effort. The PCTE is an interface set--a set of manual pages that define interfaces for UNIX. It is also two prototype implementations: a set of extensions to the UNIX kernel and the portable piggyback in Ada. The Emeraude is a French government-sponsored PCTE implementation with an emphasis on quality rather than prototyping. The British ALVEY, a program similar to the United States' STARS program, has adopted Emeraude as the underlying operating system for their software engineering environment products.

At the present time, the PCTE is not a U.S. product and there is no U.S. participation.

The project management of the PCTE is technically oriented. There are no USA-style marketing considerations. There is very little capitalization, and staffs consist of a small number of high quality people. The PCTE has many features that will probably influence the CAIS II.

The European Esprit group has also allocated money for a program called PACT. Bull has the contract for PACT which involves development of software engineering tools.

1.4 *E&V Reference Manual And E&V Guidebook*

Mr. Bard Crawford
Mr. Peter Clark
The Analytic Sciences Corporation (TASC)

Bard Crawford explained that he would be giving a general overview of the technical approach to the development of an E&V Reference System, a topic presented at the AdaJUG meeting on March 25, 1986. Peter Clark will then use some specific examples from the documents to show how various elements tie together.

Bard listed the following questions to be considered at this E&V meeting:

- What does the community want to see covered first, especially in terms of the Guidebook?

- Can/should the team do any tool/APSE cataloging?

- Shall the team discuss APSE versus IPSE?

- How does the team define E&V category 5--formal method of validation?

- Shall the Configuration Management Plan be implemented now?

These issues will be presented for discussion on Friday along with a proposed schedule for the review and update of the various documents. It is important to learn how the team views the approach taken to the documents and to determine a possible time frame for their public release.

Bard began the presentation with an outline of his projected topics: three E&V documents, E&V technologies, a conceptual model of the system, structure and uses of the system, and future plans.

The Classification Schema was written as a preliminary document to provide a framework for the Reference Manual. The Reference Manual serves as a path into the Guidebook. The two documents together are referred to as the Reference System.

A major purpose of the Reference Manual is to allow the user to find the appropriate indices for Guidebook references to the E&V technique needed to achieve a particular objective. The Guidebook is a repository for descriptions of E&V technololgy--some detailed and some synopsized with references to other documents or agencies for further information.

The Reference System is meant to provide a framework for understanding environments and how to assess them, as well as to provides specific definitions of elements and cross references between elements. This system is to be a guide to the literature of environments and assessment techniques for environments, tool sets, and tools. It will also provide detailed descriptions of specific instances of assessment technology. Assessment consists of two parts: evaluation of performance and validation of conformance to standards.

Examples of E&V technology sponsored by the team in some manner include:

- Functional Taxonomy as a checklist
- Operational Definition of the CAIS
- CAIS Validation Capability (CVC)
- IDA Benchmark programs
- Ada Compiler Evaluation Capability (ACEC)

Some E&V products that have come from other sources, but may be referenced in the Guidebook include:

- Ada Compiler Validation Capability (ACVC)
- DACS/RADC Tool Catalogue
- Ada Europe Guidelines

- Aerospace definition of a production quality compiler

- WIS tool evaluation criteria

A new conceptual diagram of the Reference Manual lists the following indices: life cycle phase; tools/tool sets/APSE; function; attribute; and E&V category. More indices may be added. The diagram indicates two ways to use the manual. One method, referred to as reducing relationships, narrows the search for information if the user just wants to read a definition or get a list of existing technology. The second method (combining relationships) leads the user to specific E&V objectives and techniques.

The Guidebook is basically structured on the five E&V categories. A primary objective is that both the Reference Manual and the Guidebook be consistent and learnable.

Bard closed his part of the presentation by presenting a list of future plans. Near-term plans (next 3-6 months) include developing cross references as a way to incorporate available products not sponsored by the E&V Team; developing the tool/APSE index; and consulting with experts in various relevant fields. Long-term goals are to consider an automated, online reference system and to expand the Guidebook entries.

A member of the audience expressed concern over the notion of trying to incorporate every product in the world that may need assessment. Bard stated that team decisions need to be made in this area.

Peter Clark began his part of the TASC presentation by listing the specific parts from each document that he would use to show interaction between the various elements.

Chapter 5 of the Reference Manual deals with the life cycle phases. There are basically six phases taken from the DoD-STD-2167 that address programming environments and software development. There are other phases relevant to whole APSE issues and a global non-phase section for those functions not specific to one phase. A text frame for a life cycle phase contains the name, description, and section number, plus cross references to functions and deliverables.

Chapter 6 addresses attributes. The text frame contains the definitional part and a section for Guidebook references, but there are no internal cross references. For a functionally-dependent attribute, the Guidebook section lists the name and section number of functions that pair up with that attribute and the assessment technology associated with those functions. A functionally-independent attribute has no pair, so the Guidebook section just references the appropriate technology.

A text frame from Chapter 7 on functions shows the type of information contained therein. An example of a function referenced life cycle phases, tools, paired attributes, and assessment technology.

Chapter 8 on tools is called a catalogue in the Reference Manual and contains sections for tools, tool sets, APSEs, and vendor agents. It will contain basically the same type of information, although there are no detailed references at this time.

An example of a use for the index was a list of deliverables and the life cycle phases pertaining to them. This index listing is in lieu of having a whole chapter on deliverables in the Reference Manual.

The E&V Guidebook has a chapter on synopses listing specific instances of technology. Chapters 5 through 9 deal with methods and information in categories such as purpose, references, vendors. They also contain details on the method itself in terms of input, output, and process.

Peter ended the presentation with some checklist and evaluation examples. Some discussion followed regarding where and how to provide vendor addresses and telephone numbers.

Wednesday's general session of the E&V Team meeting was adjourned. The remainder of the day was spent in the individual working groups.

2.0 THURSDAY, 6 MARCH 1986

2.1 APSEWG Survey

The entire team met at the request of the APSE Working Group (APSEWG) chairperson, Liz Kean, who distributed a survey meant to determine which attribute-function combinations are considered most important by various types of users.

The APSEWG is commencing an effort to establish some criteria for evaluating the ALS, the ALS/N, and the SDME by applying the REQWG attributes to the functions of these systems.

Except for this brief meeting, the entire day was devoted to individual working groups.

3.0 FRIDAY, 7 MARCH 1986

3.1 Remarks

Chairperson Ray Szymanski reopened the general session by extending thanks to everyone for attending, to Betty Wills for the name tags, and to Jimmy Williamson for handling the meeting arrangements.

3.2 Procurement Status

Mr. James Williamson

Jimmy Williamson gave a brief talk on the procurement status of the CAIS Validation Capability (CVC) and the Ada Compiler Evaluation Capability

(ACEC). Due to a delay in the original release date of the CVC, the RFPs for the CVC and the ACEC will probably be released at the same time with both efforts having a projected contract start of August 1986. A lot of people have expressed interest in these efforts, as well as in the IDA Prototype Test Suite that was released.

In answer to some questions from the audience, Jimmy explained that the Statement of Work (SOW) for the CVC is included in the RFP. It was essentially written by him and Virginia Castor and is approximately 18 pages long. After it has been released and the proposals are in, he will have copies available for the entire E&V Team.

3.3 General Business

The sixteen action items from the December E&V Meeting were reviewed. The status of those items can be found in section 3.7.

A decision was made to defer preparation of the general session minutes in order to have minutes of the working groups (SEVWG and REQWG) distributed before the Ada Europe Conference.

There are no plans for a workshop this year. In an effort to recruit new team members, a suggestion was made for a CBD announcement requesting interested persons to come in and do a position paper.

Ray Szymanski announced a tentative agenda for the June meeting which consists of the presentation by Mary Tompkins, one by Lt. Jon Wood on the Interactive Ada Workstation (IAW), a status report from TASC, and possibly a presentation by Jon Squires of the Performance Issues Working Group (PIWG).

There was a request to include the number of pages in any document that is put on the NET. Gary McKee mentioned that the APSE Component Validation Procedures Document has been on the NET since mid-January and there has been no team comment on it. Sandi Mulholland and Ronnie Martin volunteered to review the document and comment at the June meeting.

Other items of business:

- Ray was requested to review the E&V Information banners from the viewpoint of the distinguished reviewers.

- Additions to the team mailing list include Matt Emerson, Mike Mills, and Mars Gralia. New NET accounts will be opened for Mary Tompkins and Ray Sandborgh.

- Working group sections of the E&V Plan will be sent to the chairpersons for distribution to the proper people for updating.

- Bill Riddle was suggested as a potential speaker for the September E&V meeting.

- Several team members were interested in extending the E&V meetings to three full days. After some discussion, the majority favored extending the Friday session to include a general session in the morning and working groups in the afternoon. Ray Syzmanski is taking this matter under advisement.

3.4 TASC Update

Mr. Bard Crawford

Bard Crawford reported some feedback he had received on the Reference Manual and the Guidebook. It was recommended that the Reference Manual be given a more direct approach by moving the background and historical material to another document. Other suggestions involved the improvement of appearance, useability, and readability of the documents.

A Strawman review/update schedule was presented with suggested cut-off dates for team review. Some members requested more time to examine the documents, so additional input will be given at the June E&V Team meeting. A newly proposed schedule will be put on the NET at a later date. Once these documents are publicly released, there will be a major update each year.

The following points were brought out during a discussion on the questions that were put forward on Wednesday:

- Upon receipt of the Computer Sciences Corporation (CSC) material, Jerry Brookshire and Sandi Mulholland will be completing the Tools and Aids Requirements and putting it on the NET.

- During the next quarter the REQWG will be addressing the question of satisfying E&V requirements.

- The Reference Manual will continue to have a tools/APSE chapter until a decision has been made on that issue.

- There is no clear interpretation of a formal method of validation.

- The Configuration Management (CM) Plan will be implemented after a few minor corrections.

- Automation of the Reference System will remain in the concept stage until resources become available.

3.5 Working Group Status Reports

3.5.1 Coordination Working Group (COORDWG) Status Report

The COORDWG chairperson, Don Jennings, reported one personnel change. Dorothy John is a new member and will be assisting Jimmy Williamson.

Deliverables due this quarter were the E&V Status Report and the December minutes, which also comprise this quarter's accomplishments. The Status Report should be on the NET early next week. Key issues addressed included: the CM Plan; the E&V Plan, Version 3.0; and the E&V Public Report, Volume II. Because draft documents cannot appear in the Public Report, several current documents will not be included until Volume III comes out. No unresolved problems or action items were reported. Projected work for next quarter is the status report, the minutes, and Version 3.0 of the Public Coordination Strategy Document, which is also a deliverable due then.

3.5.2 APSE Working Group (APSEWG) Status Report

The APSEWG chairperson, Liz Kean, welcomed Mars Gralia back after a long absence. The deliverable due this quarter was the APSE Analysis Document which was released in January. The copy going to Europe does not contain the taxonomy because of the distribution restriction. Accomplishments this quarter included completion of the attributes survey and the SDME taxonomy. Projected work for next quarter involves taking the attributes from the survey and applying them to the ALS as a start. Of the six attributes surveyed, the one most wanted is power. Various methods and operational details will be examined next quarter. There are no deliverables due next quarter. Members took action items to consider the chosen attributes with respect to various functions.

3.5.3 Requirements Working Group (REQWG) Status Report

Pat Lawlis announced that she and Helen Romanowsky will be co-chairpersons for the REQWG until Pat leaves after next quarter to pursue a PhD. Additions to personnel were Matt Emerson (Naval Avionics Center) and Barbara Rhoads, recorder. The deliverable due this quarter was Version 2.0 of the Requirements Document, which has been finalized and will be on the NET next week. Accomplishments were completion of the Requirements Document; a SIGAda presentation; a meeting with the review of all TASC documents, including some Run Time Environment (RTE) issues for the Classification Schema; and examination of the REQWG (and E&V) scope. The scope was one of the key issues addressed this quarter and will continue to be a major topic at future meetings. Relevant topics are collectively called technology transition, and those topics addressed this quarter were the proposed information database and ways of dealing with E&V technology not generated by the team. Other key issues addressed included: review of the team with respect to its own requirements, review how team products are meeting community requirements, and ways of monitoring the use of E&V technology. There were several action items which are currently listed in the REQWG minutes. Emphasis was placed on having information put on the NET. Projected work for next quarter is the Ada Europe presentation and the technology transition issues. Helen Romanowsky will put an agenda on the NET before the next meeting listing those issues in an order for discussion. A draft of the Tools and Aids Requirements Document is the deliverable due next quarter. There are no presentations planned and no other significant information.

3.5.4 Standards Evaluation and Validation Working Group (SEVWG) Status Report

Gary McKee introduced himself as the official chairperson for the SEVWG. He acknowledged visitors, Mary Tompkins (Lookheed) and Bob Richards (EG&G Idaho) and recorder Jane Shirley. Issues addressed this quarter were the CAIS Analysis Document (CAD) and the E&V Charter. Quarterly activities were a SIGAda presentation and a Birds of a Feather session. Projected work for next quarter includes the Ada Europe presentation, Version 4.0 of the CAIS Analysis Document, and review of three TASC documents and several KIT/KITIA documents. Action items are listed in the SEVWG minutes. Tim Lindquist will give the E&V Team a briefing on the CAD at the June meeting. There are no deliverables due next quarter and no other significant information.

3.6 Closing

Ray Szymanski commended the team for their progress and thanked everyone for attending. The E&V Team meeting for March 1986 was adjourned.

APPENDIX A

ACTION ITEMS AND RESOLUTIONS
FROM THE
SEPTEMBER E&V MEETING

	ITEM	STATUS
AI-12-6-85-1	Systran. Mail presentation material to attendees.	Accomplished
AI-12-6-85-2	Systran. Put draft of December minutes on the NET.	Accomplished
AI-12-6-85-3	Systran. Prepare minutes of the REQWG and SEVWG meetings.	Accomplished
AI-12-6-85-4	Szymanski. Schedule a meeting between the ARTEWG and the E&V Team for SIGAda.	Accomplished
AI-12-6-85-5	Szymanski. Establish contact with Space Command, Colorado Springs, CO	No action
AI-12-6-85-6	Szymanski. Send thank you note to Harris Corp. for hosting.	Accomplished
AI-12-6-85-7	Szymanski. Give briefing on E&V at SIGAda.	Accomplished
AI-12-6-85-8	Szymanski. Arrange for E&V Birds of a Feather meeting at SIGAda.	Accomplished
AI-12-6-85-9	Szymanski. Track Ada Europe interest in E&V effort. Investigate E&V participation in May Ada Europe Conference.	Accomplished
AI-12-6-85-10	Szymanski. Investigate support contract capacity for additional administrative support for working groups.	Accomplished
AI-12-6-58-11	Szymanski. Find out what the "National Test Bed" is.	Dropped (no current interest)
AI-12-6-85-12	Szymanski. Develop company reaffirmation letter for REQWG members or any team member if requested.	Pending

	ITEM	STATUS
AI-12-6-85-13	Szymanski. Provide a detailed agenda for next meeting to include a list of deliverables and the status of each working group in relation to the schedule.	Accomplished
AI-12-6-85-14	Szymanski. Seek additional active members for the SEVWG.	Pending (some contacts established)
AI-12-6-85-15	Szymanski. Investigate methods of putting documents onto diskettes for Systran to format and print.	Accomplished
AI-12-6-85-16	Szymanski. Review the APSE Component Validation Procedures Document and send comments to John Reddan.	Accomplished

APPENDIX B

MARCH ACTION ITEM LIST

AI-3-7-86-1 Systran. Prepare and distribute minutes for the REQWG, SEVWG, and general session.

AI-3-7-86-2 Systran. Put E&V acronym list on the NET.

AI-3-7-86-3 Lindquist. Brief team on CAIS Analysis Document at June meeting.

AI-3-7-86-4 Mulholland and Martin. Review ACEC Document.

AI-3-7-86-5 Williamson. Have copies of the CVC SOW after its release.

AI-3-7-86-6 Szymanski. Prepare CBD announcement soliciting position papers from interested persons.

AI-3-7-86-7 Szymanski. Review E&V information banners.

AI-3-7-86-8 Szymanski. Add Matt Emerson, Mars Gralia, and Mike Mills to team mailing list. Get new accounts for Mary Tompkins and Ray Sandborgh.

AI-3-7-86-9 Szymanski. Invite Jon Squires (PIWG) to speak in June.

AI-3-7-86-10 Szymanski. Request that Ada Europe people digest E&V documents before team's arrival.

AI-3-7-86-11 Szymanski. Check with technical support contractor for possibility of generating Strawman documents.

AI-3-7-86-12 Szymanski. Send E&V documents to John Nissen.

AI-3-7-86-13 Szymanski. Investigate possibility of work product target for Ada Europe and E&V Team.

AI-3-7-86-14 Szymanski. Investigate possibility of Ada Europe coming to E&V Team.

AI-3-7-86-15 Szymanski. Check on word processing capability for team meetings.

AI-3-7-86-16 Szymanski. Investigate possibility of COORDWG's development of a 2-3 page summary of E&V activities.

APPENDIX C

ACRONYMS

ACEC . . . . Ada Compiler Evaluation Capability

ACVC . . . . Ada Compiler Validation Capability

ALS . . . . Ada Language System

ALS/N . . . . Ada Language System/Navy

APSE . . . . Ada Programming Support Environment

APSEWG . . . . APSE Working Group

CAD . . . . CAIS Analysis Document

CAIS . . . . Common APSE Interface Set

CM . . . . Configuration Management

CMU . . . . Carnegie-Mellon University

COORDWG . . . . . . Coordination Working Group

CVC . . . . CAIS Validation Capability

DACS . . . . Data Analysis Center for Software

DoD . . . . Department of Defense

E&V . . . . Evaluation and Validation

HFE . . . . Human Factors Engineering

IAW . . . . Interactive Ada Workstation

IDA . . . . Institute for Defense Analyses

IPSE . . . . Integrated Project Support Environment

LAN . . . . Local Area Network

MIT . . . . Massachusetts Institute of Technology

NAC . . . . Naval Avionics Center

PAPS . . . . Portable Ada Programming System

PCTE . . . . Portable Common Tool Environment

PIWG . . . . Performance Issues Working Group

RADC . . . . Rome Air Development Center

REQWG . . . . Requirements Working Group

RFP . . . . Request for Proposal

SDME . . . . Software Development and Maintenance Environment

SEVWG . . . . Standards Evaluation and Validation Working Group

SIGAda . . . . Special Interest Group Ada

SOW . . . . Statement of Work

STARS . . . . Software Technology for Adaptable Reliable Systems

TASC . . . . The Analytic Sciences Corporation

APPENDIX D

DOCUMENTS DISTRIBUTED
AT THE
MARCH E&V MEETING

1. Agenda for 5-7 March 1986 E&V Team meeting

2. Presentation materials used in "Human Factors Engineering (HFE)"

3. Presentation materials used in "Portable Common Tool Environment (PCTE)"

4. Presentation materials used in "Technical Approach to the Development of an E&V Reference System"

5. E&V Status Report

6. Proposed Table of Contents of E&V Public Report, Volume II

7. APSEWG Survey

8. Ada Europe Conference information package

9. Team membership and attendance list

10. E&V Reference Manual, Draft Version 1.1

11. E&V Guidebook, Draft Version 1.0

12. E&V Plan, Version 2.0

13. E&V document format

14. Minutes of December E&V Team meeting

APPENDIX E

LIST OF ATTENDEES

Brookshire, Jerry
Texas Instruments
PO Box 660246
MS 3114
Dallas, TX 75266

Clark, Peter
TASC
One Jacob Way
Reading, MA 01867

Crawford, Bard
TASC
One Jacob Way
Reading, MA 01867

Emerson, F. Matthew
NAC
Code 825
6000 E. 21st Street
Indianapolis, IN 46219

Facemire, Jeff
Texas Instruments
6550 Chase Oaks Dr.
PO Box 869305
M/S 8435
Plano, TX 75086

Fischer, Herm
Mark V Business Systems
16400 Ventura Blvd.
Encino, CA 91436

Fleming, Rich
Aerospace Corp.
M1/112
PO Box 92957
Los Angeles, CA 90009

Gicca, Greg
GTE Gov't Systems
1 Federal St.
Billerica, MA 01821

Gilroy, Kathleen
Software Productivity
Solutions
PO Box 361697
Melbourne, FL 32936

Gralia, Mars
Applied Physics Lab
Johns Hopkins University
Laurel, MD 20707

Harto, Debra
AFATL/DLCM
Eglin AFB, FL 32542-5000

Hazle, Marlene
MITRE Corp.
Burlington Rd.
Bedford, MA 01730

Jennings, Don
OC-ALC/MMECO
Tinker AFB, OK 73145-5990

John, Dorothy
AFWAL/AAAF-1
WPAFB, OH 45433

Kean, Elizabeth
RADC/COEE
Griffiss AFB, NY 13441-5700

Lawlis, Patricia
AFIT/ENC, Bldg. 640
WPAFB, OH 45433

Lindquist, Tim
Arizona State University
Tempe, Arizona 85287

Maher, Patrick
Magnavox Electronics
Systems Co.
1313 Production Rd.
Ft. Wayne, IN 46808

Martin, Ronnie
Georgia Institute of Technology
School of Information and
Computer Science
Atlanta, GA 30332

McKee, Gary P.
Martin Marietta
Information-Communications
M/S L1640, P.O. Box 179
Denver, CO 80201

Meirink, Michael
Sperry Corporation
DPG
PO Box 64525
St. Paul, MN 55164

Mills, Mike
ASD-AFALC/AXTS
WPAFB, OH 45433

Mulholland, Sandi
General Dynamics
6100 Western Pl. Suite 735
Ft. Worth, TX 76107

Richards, Robert
EG&G Idaho, Inc.
PO Box 1625
Idaho Falls, ID 83415

Rhoads, Barbara
Systran Corp.
4126 Linden Ave.
Dayton, OH 45432

Romanowsky, Helen
Rockwell International
400 Collins Rd. NE
Cedar Rapids, IA 52498

Sandborgh, Raymond
Sperry Corporation
Knowledge System Ctr.
Suite 223
3001 Metro Parkway
Bloomington, MN 55420

Shirley, Jane
Systran Corp.
4126 Linden Ave.
Dayton, OH 45432

Szymanski, Raymond
AFWAL/AAAF-2
WPAFB, OH 45433-6543

Tompkins, Mary
Lockheed Missiles
& Space Co., Inc.
2100 East St. Elmo Road
Austin, TX 78744

Williamson, Jimmy
AFWAL/AAAF-2
WPAFB, OH 45433-6543

Wills, Betty
CCSO/STA
Tinker AFB, OK 73145

APPENDIX G

MINUTES

of the

EVALUATION AND VALIDATION (E&V) TEAM MEETING

4 - 6 JUNE 1986

The task for the Evaluation & Validation of Ada* Programming Support Environments (APSEs) is sponsored by the Ada Joint Program Office (AJPO).

*Ada is a Registered Trademark of the U.S. Government (Ada Joint Program Office)

TABLE OF CONTENTS

1.0 WEDNESDAY, 4 JUNE 1986 . . . . G-3
  1.1 Welcome, Introduction, and General Business G-3
  1.2 Interactive Ada Workstation (IAW) G-3
  1.3 Ada Benchmarking G-7
  1.4 Sonicraft Experience With Ada in Weapons Systems. . . . G-8
  1.5 Program Review G-10

2.0 FRIDAY, 6 JUNE 1986 . . . . G-12
  2.1 The Analytic Sciences Corporation (TASC) Update G-12
  2.2 Working Group Status Reports G-13
    2.2.1 APSE Working Group (APSEWG) Status Report . . . G-13
    2.2.2 Coordination Working Group (COORDWG) Status Report G-14
    2.2.3 Requirements Working Group (REQWG) Status Report G-14
    2.2.4 Standards Evaluation and Validation Working Group (SEVWG) Status Report G-14
  2.3 Closing G-15

APPENDIX A ACTION ITEMS AND RESOLUTIONS FROM THE MARCH E&V MEETING G-16

APPENDIX B JUNE ACTION ITEM LIST G-18

APPENDIC C ACRONYMS G-19

APPENDIX D LIST OF DOCUMENTS DISTRIBUTED AT THE JUNE E&V MEETING G-21

APPENDIX E LIST OF ATTENDEES G-22

1.0 WEDNESDAY, 4 JUNE 1986

1.1 Welcome, Introductions, and General Business

The Evaluation and Validation (E&V) Team meeting began with opening remarks by chairperson Raymond Szymanski, followed by several announcements:

- Lt. Rick Long of AFWAL/AAAF will give a briefing on Small Business Innovative Research (SBIR) on Friday.

- The September E&V Team meeting will be held at Wright-Patterson AFB on 3-5 September 1986.

- Seven members of the E&V Team attended the Ada Europe Conference held in Edinburgh, Scotland in May.

- Ray Szymanski felt that the Ada Europe meeting was successful in several areas, the most important being the establishment of contact with a very good source of information.

- The Ada Europe Committee has recommended a continuing dialogue between the two organizations.

The following introductions were made:

- Kathy Kirkbride from SYSTRAN will be working in support of the E&V team.

- Captain Bruce Hanna, a reservist, has been assigned to Ray Szymanski's organization to work on E&V. Bruce is a graduate student at George Washington University.

- Mr. Jack Foidl from TRW is the support contractor for the KIT/KITIA and will be working with the SEVWG at this meeting. Jack is also a member of the Compliance Working Group (COMPWG).

Raymond Szymanski introduced the speakers for Wednesday's session: Captain Jon Wood and Lt. Robert Marmelstein from WPAFB, Mary Ann Tompkins from Lockheed Missiles & Space Company, Austin Division, and Fred Francl from Sonicraft.

1.2 Interactive Ada Workstation (IAW)

Captain Jon Wood & Lt. Robert Marmelstein
AFWAL/AAAF-2

The presentation given by Captain Wood and Lt. Marmelstein was entitled "Interactive Ada Workstation." The key topics for discussion included: AFWAL/AAAF projects, software productivity, workstations, and the Interactive Ada Workstation.

Projects are currently being carried out by AFWAL/AAAF in the following areas:

- Ada. The group is managing a contract to develop a cross compiler for the 1750A microprocessor.

- Embedded Computer Resources Support Improvement Program (ESIP). This environment is characterized by a modular generic simulation which can be quickly reconfigured to support a variety of fighter aircraft. The ultimate aim of this project is to eliminate any of the problems encountered while reprogramming the avionics software.

- Evaluation & Validation (E&V). The E&V Team's purpose is to oversee the development of the Ada programming support environments.

- Artificial Intelligence (AI) Research. The research being done at AAAF is targeted towards studying the human brain and its series of neural networks in order to emulate the human thought process. The ultimate goal is to produce an alternative computer architecture for implementing artificial intelligence.

- Interactive Ada Workstation (IAW). The purpose of the IAW is to increase programmer productivity while reducing software costs. This is done by providing an environment where software productivity is a function of programmer creativity and not programmer error.

It was stressed that better ways are needed to develop the increasingly complex software systems. Two possible solutions are to increase productivity in lines of code per day and to reduce errors.

There are five ways to enhance productivity.

- Automated Project Management.

- Rapid Response Time. The result of increased response time is an increase in the programmer's workability.

- Reusability of Existing Software. Software packages must be written abstractly enough that they can be reused. Reusable software is probably the most important method of increasing programmer productivity.

- Automatic Configuration Management.

- Automatic Code Generation. This refers to generating code either from specifications of written form or from diagrams. There are two approaches: 1) rapid prototyping which involves an iterative process including semi-automation, end-user participation, and benefits from actual experience, or 2) requirements driven which is characterized as one-time, fully

automatic, and by force of intellect alone.

In the efforts to increase response time, the key ingredient is the edit-compile-link-load-run cycle. This allows the programmer to sit down, edit the program and then go through the compile-link-load-run process to evaluate the program. Another element is the incremental compilation/evaluation. Instead of using a batch compilation process, the incremental compilation/evaluation method uses a process whereby a programmer can write a fragment of a program and see the results of that program immediately. This decreases response time which, in turn, will increase productivity.

The Intelligent Workstation was presented as one of the more important advances in computer technology in recent years. It is characterized by a highly specialized environment which will aid in work completion. This is accomplished by placing a variety of design and analytical tools at the user's disposal. The capabilities of these next generation workstations will be targeted primarily to the following areas: computer aided engineering, automatic code generation, expert system tools, and interactive instruction and analysis. The emergence of several new technologies has served to set this next generation workstation apart from its predecessors in terms of power, performance, and price. Examples are the mouse cursor control and high resolution graphic displays, large main memories on the order of 10 megabytes, expert systems, and Ada which will enhance the portability of software between workstations.

The Symbolics 3600 was chosen as the host for the Interactive Ada Work-station. It provides many tools for constructing an extremely good Ada programming support environment. These features include: a language-oriented source code editor, an interpreter loop for interactive execution, a window-based debugger and inspecter, and system maintenance facilities. There is no security system which means the user may redefine the level of workstation functions.

The IAW core system components are the host environment support, program development tools (these are aimed toward automatic code generation), and program support tools.

The program development tools are composed of four editors: Buhr diagram editor, syntax directed editor, state diagram editor, and the hot editor. The Buhr diagram editor was taken from the book SYSTEM DESIGN WITH ADA by Ray Buhr, and it expresses high level system interfaces. It is hierarchical and does not express low level control flow. The Buhr diagram editor is composed of an object, a task, a package, and a connection.

The state diagram editor is used by compiler designers in building top down cursor distinct compilers. It expresses low level control and is composed of two items, states and transitions.

The hot editor is a combination of multiple views of code that allow the programmer to instantly determine the results of a given modification.

Other program support features of the IAW include the following:

- Command Processor/Interpreter. This is a user interface interpreter. The interpreter executes LISP macros on a Symbolics 3600 Interactive Interpreter. These LISP macros emulate Ada statements and make up an intermediate language called IAda.

- Smart Help System. This is an application of expert system technology that is geared toward helping the user select the proper commands for a given operation, and will also give information on how to use those commands.

- Expert System Tools. This includes an expert system produced by General Electric to help the user debug his IAda program.

- Entity Manager. This is the most important component of the Interactive Ada Workstation and has a dual purpose. One purpose is to keep track of local objects. This is important especially for implementing the hot editor and the incremental compilation. The second purpose is to keep track of the global data base components. These components are documentation such as test libraries, help files, and test data.

- Project Management Tools. These tools include an entity manager that tracks the data base for every particular project that the user would want to implement on the Interactive Ada Workstation.

- Smart Librarian. The Smart Librarian will be an expert system targeted toward locating reusable software components in the memory or in the IAW data base. It will do this by taking an incomplete specification from the user on the algorithm to be looked up, and searching through the data base, based on that specification, prompting the user for additional information as needed.

- Self-Documenting Code Specification. The editors used to produce the Ada program, such as the Buhr diagram editor and the state machine editor, provide a convenient method for specifying the high level flow charts of the diagram and the individual algorithm specifications for each module, thereby presenting self-documenting code specification as a result.

The following points were brought out during a question and answer period that concluded the presentation:

- The smart librarian is not being worked on at all right now due to financial problems.

- The question of "who owns what" in software is a cloudy issue. In this particular case, there is an understanding that this software will be distributable.

- This is a research-oriented project. It is not necessary that a final product emerge from it.

- The state diagram editor is fully working, as is the Buhr diagram editor. However, there are errors in them; they don't generate code perfectly.

- This project does not currently have an ITAR restriction.

1.3 Ada Benchmarking

Mary Ann Tompkins
Lockheed Missiles and Space Company

The major topics of Ms. Tompkins' presentation were the methodology, the taxonomy, and the development process used in building a benchmark.

In methodology, the features are modularity, time measurement, portability, code generators, and a common interface. The modularity of benchmarking involves the ability of selection. Each benchmark measures or evaluates only one feature at a time. Only those benchmarks that evaluate the desired feature can be selected.

Time measurement is done by clock time. There is a controversy over whether to use central processing unit (CPU) time or clock time. Everyone doesn't report CPU time in the same manner, but most people can determine a time interval using a wall clock.

Portability uses Ada attributes and code to detect dependencies. Code generators provide the ability to evaluate an attribute at a particular limit, and are useful for implementing specific features.

The methodology also involves a common interface. With this, the user can define parameters such as the number of iterations, output files, and the size of an object. With the common interface, the benchmark is a reusable package that is generic. It can be instantiated with the commands that are legal for that one particular benchmark.

The taxonomy of Ada benchmarking is divided into four categories: sample program, major feature tester, small scale feature tester, and prototype building block/utility. The sample program contains algorithms typical in real-time systems and demonstrates the

operation of the environment. It is a general system tester.

The major feature tester is composed of task activation, task rendezvous, task scheduling and exception handling. The major feature tester evaluates the overhead of a feature and demonstrates its implementation. It also impacts design decisions.

The small scale feature tester is designed to test small scale features such as assignment statements and procedure calls. The tester evaluates the performance of the code generated by the compiler. In the tuning phase, implementation decisions of certain algorithms might be impacted.

In the prototype building block/utility, the prototype building blocks are modules that may be used to build multiple benchmarks. The utilities are programs that promote information gathering, such as Ada source/assembly code counters.

The presentation concluded with a discussion of the development process. The need for a description/requirements document was discussed. This document covers the following areas: description, classification, attributes measured, measurement, machine dependence, transportability, required language proficiency, evaluation and commands.

The need for portability was stressed. The benchmarks are developed on a VAX, and the source code is then ported to Data General and Rational to verify the design, implementation, and portability.

At this point, Mary Ann ended her presentation by answering some questions from the audience relating to the compilation speed of various machines, target timing, incremental compilation, and future areas of development.

1.4 Sonicraft Experience with Ada in Weapons Systems

Fred Francl
Sonicraft

The presentation began with some brief comments on Mr. Francl's background. He has been in management for 20 years, not all of it software management. His duties at Sonicraft involved implementing an Ada weapons system project. The focus of the presentation was Sonicraft's experience with Ada.

Sonicraft was tasked to marry Ada to JOVIAL J73. Significant advantages were seen in going with Ada, not the least of which was that JOVIAL was on the way out and Ada was on the way in. It was also felt that the Ada language capability was far superior to JOVIAL. The choice was between being one of the last people to use JOVIAL or one of the first to use Ada, and this was one of the major factors in Sonicraft's decision to use Ada.

The Ada Language System (ALS) was to be retargeted for Sonicraft's application, and this caused inefficiencies to be built into it. A year's slack had to be built into the schedule to provide for problems in the compiler development. Sonicraft built on the ALS, but even before they were under contract, there was a one year slip in the ALS delivery. Another slip occurred when the Ada standard was changed to meet the American National Standards Institute (ANSI) requirements.

When the Sonicraft C-5 software development did not procede as was hoped, the customer became alarmed. There was much customer attention as a result, and Sonicraft had to hire several consultants to help handle the intensive review cycle. The C-5 software was developed using Ada program design language (PDL) and was ready in time for the Critical Design Review (CDR). A problem of time loss occurred when the ALS version was found inadequate. Consequently, the CDR demonstrations were performed using a combination of JANIS, Ada, Pascal, and some assembly.

With the task manager, the first problem was related to sizing. A base for each possible interrupt that could occur had to be allocated for each task. One of the first steps taken was to restrict the types of interrupts that could be accepted. This immediately reduced the memory requirements. Another step was to decrease times when no interrupts could be enabled because something else was being processed. One method of achieving this was to implement assembly language code as an alternative to the code generated by the compiler. One of the biggest problems in this particular system was that the priorities of tasks were set at compile time. Many alternative methods were explored including looking at different sets of tasks which could have different priorities even though they were performing the same function. The interrupts were the only task manager items that were looked at since interrupts are required to be handled as tasks.

When storage is needed, it must be contiguous. Once the storage becomes fragmented, a contiguous block can no longer be found for whatever is needed to be done. One way to make sure the memory stays contiguous is to put in garbage collection routines which will reassign memory. In studies conducted by Sonicraft, the NEW command was found to cause memory fragmentation; therefore, the use of this command was restricted in their coding standards.

Because there are many restrictions in microprocessor application, the objective is to get as much as possible in the life cycle cost benefits of a project.

A unique set of diagnostic tools is needed to do high level language development. As a result of a trade study, tracing was accepted only at the task level. Tracing could be done within a task, but if a rendezvous occurred with another task, tracing was lost. Another limitation was when the debug option was selected, tracing could not be done. A way is needed to pick up built-in (BIT) tests as

part of the diagnostics. The other problem was that if BIT went to test blocks of RAM, it could only test it for hardware chips.

Again, a major factor is size. Ada is almost automatically excluded from many microprocessor applications because the Run-time Support Library (RSL) is so large. A program is going to have cost, if there are sizing problems. There are ways of optimizing sizing in the following areas: 1) in the application part of the code, the number of tasks can be cut, 2) the Run-time Support Library can be segmented to prevent linking into routines associated with one specific Ada feature, if that feature were to be restricted for any reason, and 3) the code generator within the compiler will produce the largest gain in size optimization. This gain will occur in the size of the RSL and in the object code generated.

In his concluding remarks, Mr. Francl spoke on the benefits Sonicraft had derived from their Ada experience. Sonicraft was one of the first to have any experience with Ada PDL. The C-5 software development PDL was from Softech, and there was a problem with it because it was not compilable. Thus, the interfaces could not be checked using the machine, and the first version of the specification was rejected by the customer due to interface errors. The lesson learned was not to depend on the PDL to check interface problems. In the version of the specification that was accepted, the interfaces of the PDL were coded in actual Ada code. The machine then found all the problems, and they were fixed. Therefore, using a compilable PDL is highly recommended.

The following points were brought out during a question and answer period that concluded the presentation:

- The material presented has not been published but is in draft form.

- Sonicraft did not have any IR&D funding.

- The tools were based on existing tools which like the debugging system was based on the INTEL decoder. The syntax was changed and a limitation was placed on the amount of tasking used.

1.5 Program Review

LCDR Philip Myers
AJPO

The Wednesday afternoon general business session reconvened with a program review by LCDR Philip Myers. LCDR Myers opened his briefing with commentary on the various activities in the Ada community and the increasing momentum of the Ada program.

All programs are under a major review. The Ada effort has been reviewed and the international activities, the E&V team, and the

KAPSE Interface Team (KIT) will be reviewed after the next meeting. There has been a major program review on "where we're at, where we're going, and why."

The impression must be given to the customer, which is primarily the Department of Defense Program Management, that their needs are being addressed through policies and activities. At the last Tri-Service Managers Review, all three program managers gave advice and counsel to AJPO to move out in areas of evaluation and performance. Feedback is coming in that the early users of Ada are having problems due to quality, performance, etc.

The comments on the new validation policies have been received, and they are now being reviewed.

Some decisions were made on the modification of the current compiler revalidation procedures. There will be a twelve month validation cycle. The next version of the ACVC, Version 1.8, will be released on 1 June 1986. The following one, Version 1.9, will be released on 1 December 1986 for a six month review period prior to becoming effective on 1 June 1987.

Tricia Oberndorf and the CAIS Working Group (CAISWG), part of KAPSE Interface Team/KAPSE Interface Team Industry & Academia (KIT/KITIA), are in the process of answering the more than 650 comments received during the formal MIL-STD coordination of Common APSE Interface Set (CAIS) Version 1.4. They hope to finish reviewing the comments by the July KIT/KITIA meeting.

As mentioned in the Ada Information Clearinghouse newsletter, Virginia Castor is going to chair the third meeting of the International Special Group in Ada Programming Support Environments. A statement of intent was signed by ten North Atlantic Treaty Organization (NATO) countries in cooperative efforts under the NUNN amendment initiative to do some work on Ada programming support environments. The United States has offered to do prototyping of and tool building on a subset of CAIS Version 1. This may mean that the AJPO will receive some NUNN amendment money to accomplish this task.

The following points were made during the question and answer period that concluded the presentation:

- The EV-INFORMATION directory is no longer a publicly accessible directory on Ada 20. All public address accounts have been disabled. No one on the team should have a problem since everyone has accounts on Ada 20.

- Many accounts are still in jeopardy due to not receiving any funding from the STARS program.

- The ALS Version 3 will soon be released to the public. If the Army does not continue supporting the ALS the Navy certainly will.

- At this point, Gramm-Rudman has only recouped the 1985 money and accessed the 1986 money. E&V is not affected right now.

- Every aspect of DoD is now involved in Ada. The Army has mandated information to use Ada only. When invented, Ada was intended only for mission critical systems, but it has been mandated for many other diverse functions as well. There is breadth but no depth. Depth is the function of time, and everybody in the Special Interest Group Ada (SIGAda) community and the Ada community needs to understand that it takes time. It will take a few more years, and a strong central management control of the language. What is hindering Ada is all the problems that have been around in other languages for the last 20 years. The problems are just more apparent since everyone is now speaking the same language.

- The following information was given concerning the proposal to the NUNN amendment: In the white paper, the United States proposed that CAIS Version 1 be used as a basis for cooperative Ada Programming Support Environment (APSE) work under the divisions of the NUNN amendment public law, 99-145, 8 November 1985. The provisions of the NUNN amendment were developed to promote cooperation between NATO nations in research and development on defense equipment and munitions production. Efforts would include the development of a set of software tools of which an APSE would be comprised. Implementation of these tools on a common set of interfaces upon two distinct computer architectures as well as the development of an APSE evaluation technology consistent with this developed set of tools is also expected.

2.0 FRIDAY, 6 JUNE 1986

2.1 The Analytic Sciences Corporation (TASC) Update

Peter Clark

Mr. Clark's presentation focused on an overview of TASC's efforts and goals. The key points were the proposed schedule for the documents, the Reference System issues, the Reference Manual, the Guidebook, and the Configuration Management (CM) plan that has been implemented.

A key area is the structure and organization of the documents. One suggestion was to reduce the historical background in both the Reference Manual and the Guidebook, but particularly in the Reference Manual. Another point was to focus on the whole APSE issues.

To increase the readability of the Reference System, the suggestion was made to have more figures and tables, and to take each separate section of the Reference Manual and the Guidebook and have them on separate pages. The use of multiple types, sizes, and styles would emphasize key words or focus on certain items on a page. There are a

number of lists in the cross references and the references to the Guidebook that if aligned will help legibility.

There was concern over Chapter 8 of the Reference Manual which was not complete in the manual, but dealt basically with tools in the APSE catalog. The idea was to list all the tools, each instance of a compiler, and point to vendors and features.

Another suggested approach was to define generic tools such as compilers and continue from that point. This would basically cut off work in the Reference Manual that deals with tool components.

Another suggestion was that the Data Analysis Center for Software (DAC) tool catalog has a functional taxonomy which would prove helpful if coordinated with the E&V taxonomy. If both of these systems were automated in the same host, the public could easily access both the E&V technology and the tools catalog.

One of the Reference Manual issues is the organization of the functional taxonomy. There is a need for a clear distinction between the functional dependent attributes and the functional independent attributes. The taxonomy used in the Reference Manual is hierarchical in nature and a derivative of the Software Engineering Environments (SEE) taxonomy.

Concerning the Guidebook, the most important question was how the E&V technology was organized. Currently it is organized by E&V category. Some suggestions were to order the technology by function so that all the technology for evaluating or accessing a compiler with a compilation function would be together. Another was to organize the technology by attributes.

2.2 Working Group Status Reports

2.2.1 APSE Working Group (APSEWG) Status Report

The APSEWG representative, Greg Gicca, reported that most of their members were absent. The accomplishments for the quarter were the completion of a lower level version matrix of the taxonomy vs. attributes to further evaluate the importance of certain attributes, and the correlation of major attributes vs. transformation to the E&V Reference Manual. There were no deliverables due this quarter. The key issues discussed were: to look into various APSE information gathering techniques; to develop a plan to invite various vendors to present their products to the team; to draft a letter to send to vendors which has to be finalized and submitted for approval; and to review the APSEWG charter. An unresolved problem was if the ITAR restriction on the APSEWG taxonomy could be removed. The action items include completing the draft plan/letter and submitting it for approval, and checking on the KIT/KITIA survey contents and results. There are no deliverables for next quarter. The projected work for next quarter includes: preparations to invite

vendors via the plan, if approved; completing the taxonomy descriptions of the Software Development and Maintenance Environment (SDME), if the SDME system specification is approved; and working on attribute vs. taxonomy on SDME, if the SDME system specification is approved.

2.2.2 Coordination Working Group (COORDWG) Status Report

The COORDWG chairperson, Don Jennings, reported the key issues addressed this meeting were the draft of the Public Coordination Strategy Document, Version 3; the draft of the Technical Coordination Strategy Document, Version 3; the E&V status report; and the March minutes. The latter two were also the accomplishments for the quarter. There were no unresolved problems or action items. The projected work for next quarter and the deliverables due are the status report, the minutes, and the Technical Coordination Strategy Document, Version 3.

2.2.3 Requirements Working Group (REQWG) Status Report

Helen Romanowsky, chairperson of REQWG, gave the following status report. A deliverable due this quarter was Version 2.0 of the Requirements Document. The final form of this document will be given to Ray for release. Accomplishments this quarter included the Ada Europe presentation; updates on REQWG-pertinent items in the E&V Plan; a review of the requirements to see how they are being met (the majority have been or are being addressed by some sort of action within the team); and a white paper written by Ronnie Martin titled "The Treatment of Externally Developed E&V Technology" which concerns the issue of how to incorporate into future documents technology not developed by the E&V Team. Key issues addressed this quarter: 1) the concept of a global team repository of terminology, 2) the idea of E&V user scenarios to help determine how pertinent the technology being studied will be to the user, 3) issues concerning technology not generated by the team (subject of Ronnie Martin's white paper), and 4) how the team is meeting its own requirements. The projected work for next quarter is to review: 1) the user scenarios, 2) the updated draft of the Tools and Aids Document, and 3) any unresolved comments and the European comments on the Requirements Document. This status report, Ronnie's white paper, and the latest version of the draft Tools and Aids Document will be put on the NET before the September E&V Team meeting. The REQWG has no deliverables due next quarter, no presentations planned for the next meeting, and no other significant information to report.

2.2.4 Standards Evaluation and Validation Working Group (SEVWG) Status Report

Gary McKee, SEVWG chairperson, acknowledged visitors. Jack Foidl (TRW; KIT/KITIA Support Contractor), and LCDR Philip Myers (Navy Deputy-AJPO). The key issues addressed this quarter were a review of CAIS Analysis Document (CAD), Version 3.2; a review of E&V

Classification Schema; a discussion of KIT/KITIA Compliance Working Group (COMPWG) activities; and a discussion of long term future plans. There were no deliverables or unresolved action items. "Courtesy" items (action items with unforeseeable conclusions) include: transmitting CAIS access control issues to KIT/KITIA, and recommending necessary re-wording of the access control sections of MIL-STD-CAIS. Action items for next quarter include: schedule a team review of the CAD in September; edit, review, and update CAD by 1 August 86; distribute CAD to team, establish regular NET communications with KIT/KITIA COMPWG; provide formal closure to comments from European Environments Working Group; review and edit E&V Plan in regards to SEVWG; and deliver the trip report to Ray Szymanski on Ada Europe.

2.3 Closing

Ray Szymanski met with the Working Group chairpersons. He expressed the need to reschedule the joint meeting between the E&V Team and the ARTEWG which was to take place at the Pittsburgh SIGAda meeting. Ray thanked everyone for attending. The floor was then open for general discussion.

A discussion ensued on the amount of time to spend in open discussion of the CAIS document. A block of time will be set aside for this discussion with Ray determining the length of time.

Gary McKee was questioned on what other standards he will study. Gary stated his interest in the Ada Compiler Evaluation Capability, the Graphical Kernel Support (GKS) standard, and the ARTEWG standard. The result of the ACEC study will be an ACEC evaluation document. The primary focus of the study is to evaluate actual implementations.

The group discussed the background and origin of the Ada Board along with its current status.

It was stated that getting the ITAR restriction lifted from the new taxonomy is the responsiblity of RADC who originated the restriction. This task is presently being worked on.

The E&V Team Meeting adjourned with the viewing of the Interactive Ada Workstation video.

APPENDIX A
ACTION ITEMS AND RESOLUTIONS
FROM THE
MARCH E&V MEETING

	ITEM	STATUS
AI-3-7-86-1	SYSTRAN. Prepare and distribute minutes for the REQWG, SEVWG, and general session.	Accomplished
AI-3-7-86-2	SYSTRAN. Put E&V acronym list on the NET.	Pending
AI-3-7-86-3	Lindquist. Brief team on CAIS Analysis Document at June meeting.	Pending
AI-3-7-86-4	Mulholland and Martin. Review ACEC Document.	Pending
AI-3-7-86-5	Williamson. Have copies of the CVC SOW after its release.	Pending
AI-3-7-86-6	Szymanski. Prepare CBD announcement soliciting position papers from interested people.	Cancelled
AI-3-7-86-7	Szymanski. Review E&V information banners.	Accomplished
AI-3-7-86-8	Szymanski. Add Matt Emerson, Mars Gralia, and Mike Mills to team mailing list. Get new accounts for Mary Tompkins and Ray Sandborgh.	Accomplished
AI-3-7-86-9	Szymanski. Invite Jon Squires (PIWG) to speak in June.	Accomplished
AI-3-7-86-10	Szymanski. Request that Ada Europe people digest E&V documents before team's arrival.	Accomplished
AI-3-7-86-11	Szymanski. Check with technical support contractor for possibility of generating Strawman documents.	Accomplished
AI-3-7-86-12	Szymanski. Send E&V documents to John Nissen.	Accomplished
AI-3-7-86-13	Szymanski. Investigate possibility of work product target for Ada Europe and E&V Team.	Accomplished

AI-3-7-86-14	Szymanski. Investigate possibility of Ada Europe coming to E&V Team.	Accomplished
AI-3-7-86-15	Szymanski. Check on word processing capability for team meetings.	Pending
AI-3-7-86=16	Szymanski. Investigate possibility of COORDWG's development of a 2-3 page summary of E&V activities.	Pending

APPENDIX B
JUNE ACTION ITEM LIST

AI-6-6-86-1 Szymanski. Contact ARTEWG about July meeting.

APPENDIX C
ACRONYMS

ACEC(Prototype)	Ada Compiler Evaluation Capability
AFWAL	Air Force Wright Aeronautical Laboratories
AJPO	Ada Joint Program Office
ALS	Ada Language System (Army)
ANSI	American National Standards Institute
APSE	Ada Programming Support Environment
APSEWG	APSE Working Group
ARTEWG	Ada Run Time Environment Working Group
CAD	CAIS Analysis Document
CAIS	Common APSE Interface Set
CAISWG	CAIS Working Group
CDR	Critical Design Review
COMPWG	Compliance Working Group
COORDWG	Coordination Working Group
CM	Configuration Management
CPU	Central Processing Unit
DACS	Data Analysis Center for Software
E&V	Evaluation and Validation
ESIP	Embedded Computer Resources Support Improvement Program
GKS	Graphics Kernel Support
IAW	Interactive Ada Workstation
KIT/KITIA	KAPSE Interface Team/KAPSE Interface Team Industry & Academia
NATO	North Atlantic Treaty Organization
PDL	Program Design Language

RADC	. . . .	Rome Air Development Center
REQWG	. . . .	Requirements Working Group
SBIR	. . . .	Small Business Innovative Research
SDME	. . . .	Software Development and Maintenance Environment
SEE	. . . .	Software Engineering Environment
SEVWG	. . . .	Standards Evaluation and Validation Working Group
SIGAda	. . . .	Special Interest Group Ada
STARS	. . . .	Software Technology for Adaptable Reliable Systems
TASC	. . . .	The Analytic Sciences Corporation

APPENDIX D
DOCUMENTS DISTRIBUTED AT MEETING

1. Agenda for 4-6 June 1986 E&V Team meeting.

2. E&V attendance/membership list.

3. E&V Status Report.

4. Minutes of the March E&V Team meeting.

5. Presentation materials used in "Ada Interactive Workstations."

6. Presentation materials used in "Ada Benchmarking."

7. Presentation materials used in "Sonicraft Experience with Ada in Weapons System."

8. Presentation materials used in "E&V Technical Support Activities: Report To E&V Team Meeting."

9. List of Ada Board Members.

10. Thesis by David B. Crane entitled "Requirements Analysis for the Interactive Ada Workstation (IAW)."

11. List of E&V Acronyms.

12. Raymond Szymanski's briefing at the Pentagon and at Ada Europe.

APPENDIX E
LIST OF ATTENDEES

Brookshire, Jerry
Texas Instruments
P.O. Box 660246
MS 3114
Dallas, TX 75266

Crawford, Bard
TASC
One Jacob Way
Reading, MA 01867

Foidl, Jack
TRW
9265 Sky Park Court
San Diego, CA 92123-4213

Gicca, Greg
GTE Gov't Systems
1 Federal St.
Billerita, MA 01821

Hanna, Capt. Bruce F.
6210 Pioneer Dr.
Springfield, VA 22190

Hazle, Marlene
MITRE Corp.
Burlington Rd.
Bedford, MA 01730

John, Dorothy
AFWAL/AAAF-1
WPAFB, OH 45433

Lawlis, Patricia Maj.
AFIT/ENC
Wright-Patterson AFB
OH 45433

Lindquist, Tim
Computer Science Dept.
Arizona State University
Tempe, Arizona 85287

Clark, Peter
TASC
One Jacob Way
Reading, MA 01867

Facemire, Jeff
Texas Instruments
6550 Chase Oaks Drive
P.O. Box 869305
M/S 8435
Plano, TX 75086

Francl, Fred
Sonicraft, Inc.
1401 E. 95th St.
Chicago, IL 60619

Gralia, Mars
Applied Physics Lab
John Hopkins Univ.
Laurel, MD 20707

Harto, Debra
AFTAL/FXG
Eglin AFB, FL 32542-5434

Jennings, Don
OC-ALC/MMECO
Tinker AFB, OK 73145-5990

Kirkbride, Kathy
SYSTRAN Corporation
4126 Linden Ave.
Dayton, OH 45432

Leavitt, Thomas
Boeing Military
Airplane Company
P.O. Box 7730
MS K75-67
Wichita, KS 67227

Maher, Patrick
Magnavox Electronics Systems
1313 Production Rd.
Ft. Wayne, IN 46808

Martin, Ronnie
Georgia Institute of Technology
Software Engineering Research Center
Atlanta, GA 30332-0280

McKee, Gary P.
Martin Marietta
Info & Communication
M/S L1640 P.O. Box 179
Denver, CO 80201

Myers, LCDR Philip
AJPO
Rm 3D139
(Fern St/C107)
The Pentagon
Washington DC 20301-3081

Mills, Mike
ASD-AFALC/AXTS
Wright-Patterson AFB
OH 45433-6503

Reddan, John
SYSCON Corporation
3990 Sherman St.
San Diego, CA 92110

Romanowsky, Helen
Rockwell International
400 Collins Rd NE
Cedar Rapids, IA 52498

Rhoads, Barbara
SYSTRAN Corporation
4126 Linden Ave.
Dayton, OH 45432

Shirley, Jane
SYSTRAN Corporation
4126 Linden Ave.
Dayton, OH 45432

Szymanski, Raymond
AFWAL/AAAF
Wright-Patterson AFB
OH 45433-6543

Tompkins, Mary
Lockheed Austin Division
2100 E. St. Elmo Rd.
Austin, TX 78744

Weiderman, Nelson
Software Eng. Inst.
Carnegie Mellon
Pittsburgh, PA 15213

Wills, Betty
CCSO/XPTB
Tinker AFB, OK 73145

APPENDIX H

MINUTES

of the

EVALUATION AND VALIDATION (E&V) TEAM MEETING

3 - 5 SEPTEMBER 1986

The task for the Evaluation & Validation of Ada* Programming Support Environments (APSEs) is sponsored by the Ada Joint Program Office (AJPO).

*Ada is a Registered Trademark of the U.S. Government (Ada Joint Program Office).

TABLE OF CONTENTS

SECTION	PAGE
1.0 WEDNESDAY, 3 SEPTEMBER 1986	H-3
1.1 Welcome, Introduction, and General Business	H-3
1.2 Common APSE Interface Set (CAIS) Operational Definition Update	H-3
1.3 Presentation/Discussion of the CAIS Analysis Document	H-7
1.4 The Performance Issues Working Group (PIWG)	H-9
2.0 THURSDAY, 4 SEPTEMBER 1986	H-11
2.1 Announcements	H-11
3.0 FRIDAY, 5 SEPTEMBER 1986	H-12
3.1 TASC Update	H-12
3.2 Working Group Reports	H-14
3.2.1 Standards Evaluation and Validation Working Group (SEVWG) Status Report	H-14
3.2.2 Requirements Working Group (REQWG) Status Report	H-14
3.2.3 Coordination Working Group (COORDWG) Status Report	H-15
3.2.4 APSE Working Group (APSEWG) Status Report	H-15
3.3 AJPO Update	H-15
3.4 Closing	H-16
APPENDIX A Action Items and Resolutions from the June E&V Meeting	H-17
APPENDIX B September Action Item List	H-18
APPENDIX C ACRONYMS	H-19
APPENDIX D DOCUMENTS DISTRIBUTED AT MEETING	H-20
APPENDIX E LIST OF ATTENDEES	H-21

1.0 WEDNESDAY, 3 SEPTEMBER 1986

1.1 Welcome, Introductions, and General Business

The Evaluation and Validation (E&V) Team meeting began with opening remarks by chairperson Raymond Szymanski, followed by the introductions of people:

- Capt. Bruce Pickart, Air Force Operational Test Center, Kirtland AFB, New Mexico.
- Suzanne Menichiello, Aerospace Corporation, Los Angeles, California.
- Fred Francl, Sonicraft, Inc., Chicago, Illinois.

It was announced that:

- There will be no meeting in Stockholm.
- Bard Crawford announced that hotel rooms have been reserved for the next E&V Team meeting, December 3 - 5, in San Diego. Further information will follow on the MIL-NET.

Ray Szymanski introduced the speakers for Wednesday's session: Dr. Tim Lindquist from Arizona State University, Gary McKee of the Standards Evaluation and Validation Working Group (SEVWG), and Jon Squire from Westinghouse.

1.2 Common APSE Interface Set (CAIS) Operational Definition Update

Dr. Tim Lindquist
Arizona State University

The first speaker of the Wednesday session was Dr. Tim Lindquist, a professor at Arizona State University. His presentation was titled "The CAIS Operational Definition (CAIS OD)." The objectives of this project are to create an operational semantic definition of the CAIS which is written primarily in Ada, and to continue research in APSE validation issues.

The CAIS Node_Model is composed of four types of nodes: structure nodes, process nodes, file nodes, and queue nodes. The nodes are connected by means of relationships, which are characterized by a relation name and a relationship key. A relation name indicates mapping between nodes, and the relationship keys designate unique elements of that mapping. A relationship can also be characterized by an attribute, thus enabling the characteristics of a relationship to be described. Nodes may have attributes, and the relationship between nodes can have attributes. In the CAIS, the two types of relationships are primary and secondary. A primary relationship enforces the hierarchical structure of the CAIS, composing a tree structure. The goal is to define a node

structure that will accurately depict the types of entities which can be manipulated by software engineering tools.

The basis of the CAIS operations is CAIS List_Utilities, which defines a set of routines used for manipulating the various types of lists within the CAIS. There are three types of lists: an empty list, an unnamed lists and a named list. Items that can be placed in lists include strings, integers,floats, identifiers, and sublists. The CAIS contains operations that allow the user to create and manipulate these lists.

Facilities provided within the node_management and access_control packages allow the user to create and manipulate nodes regardless of their contents or type. Types of nodes include structure nodes, process nodes, and various kinds of file nodes. All nodes have contents except structure nodes,which are used as directory nodes.

Path names are used in the CAIS Node_Model to specify a path through the graph structure using a relation name and a relationship key. The relation-relationship key pairs can be linked together forming a pathname allowing the traversal of several different nodes or several different directed arcs through the structure.

Special facilities in the Process_Control consist of spawning, invoke, create job, determine status, append/get results, get parameters, and abort/suspend/resume.

Input and output facilities are composed of four categories: (1) secondary storage, which includes sequential and direct (2) Text_I/O queues (3) terminals, and (4) magnetic tapes. Queue files provide communication between different processes. The current version of the CAIS handles scroll, page, and form terminals with text input and output. The final section, magnetic tapes, deals with text input/output.

It was noted that the Node_Model is a level of abstraction higher than software tools have encountered in the past. Development of the CAIS OD took place for three reasons: to provide a complete executing version of the CAIS to be used for tool transportability studies, to examine CAIS functionality, and to perform tool retargetability studies. The OD will also provide input to the development of the CAIS Validation Capability. The items involved here are developing validation tests, identifying and resolving specification gaps, and operationally testing validation tests. The third point is to provide the next step in a sequence of more formal specifications of the CAIS. Dr. Lindquist cited the completion status of following items. List Utilities code has been completely developed and between 450 and 550 tests run, CAIS private routines include code, testing, and documentation, the coding, testing, and documentation has been completed for Node Management. In the area of attributes, the coding and tests are completed. Access mechanism coding is completed and integration is in progress. Efforts are progressing to integrate Process Control with the rest of the CAIS in

terms of the discretionary access model. Dr. Lindquist explored a master's thesis written by David Barlow. Mr. Barlow analyzed the Process Control from the point of view of the Command Language Interpreter (CLI). His goals were to evaluate the ability of the CAIS to effectively support APSE tools, to evaluate the UNIX C Shell as a Command Language Interpreter, and to design and implement selected features of a CLI for the APSE. These features include a parse command line, syntactical analysis command, invoking a CLI tool, redirect input/output, execution in foreground/background, and generating pipelines. In the implementation of the CLI, there are four overlapping phases in terms of development: developing design procedures for logging on to the APSE; establishing syntax for the CLI and building the parser and syntax analyzer; coding the procedures that utilize CAIS functions to implement the CLI features; and implementing tools necessary to demonstrate the functions of the CLI. The first two phases were emphasized in the presentation. The first phase involves six steps: initializing a CAIS hierarchy, establishing the system manager, spawning the login monitor, spawning the login process, creating the user's root process node, and linking the user's top level node to the system manager's directory. Phase two involves syntax, parsing, and analysis. Syntax includes the BMF definition, and the variations between ASH and CSH which are one line input, no parenthesis, separate flag and string arguments, and CAIS pathnames. Parsing includes 'scan to tokens', and queues of valid Ada identifiers and CAIS pathnames. Syntactical analysis involves RRIPLL, command list, sublists, and commands. In his conclusion, Mr. Barlow noted as a CAIS deficiency the inability of the input/output to trap and respond to user-generated interrupts. He offered suggestions for CAIS enhancements in the areas of:

- Node handles as list elements
- Stack operations on lists
- Root path name
- Return node handle to queue file node
- Add output node parameter to Spawn Process and Invoke Process
- Return node handle to root process node from create job.

In commenting upon the CLI enhancements, Mr. Barlow suggested improving the ability to change and add new features; adding a command programming language; having concurrent syntax analysis; and improving response time.

Mr. Barlow's recommendations include:

- Correcting the noted CAIS deficiency
- Adding the suggested enhancements

- Completing the Operational Definition

- Adopting the CAIS as the standard Ada interface set

- Researching hardware implementation of the CAIS

- And continuing development of an integrated database as a key part of the APSE.

Another thesis presented was titled "Automated Generation of Input/Output Pairs For The CAIS Validation Test Suite" by Joyce Rene Jenkins. Ms. Jenkins provided an analysis of an implementation of the generation of input and output pairs from the operational definition in Ada code. The problem addressed was showing the consistent and complete correspondence between an implementation and a specification.

Three validation sets were presented by Dr. Lindquist. The third set is the one used; it involves the specification inside the implementation. In stating the reasons for this automation, Dr. Lindquist stressed the progress that is being made in the selection method of the best inputs and in determining the quality of the inputs. Another reason for the automation encompasses applications beyond the CAIS including software interfaces and reusability, Ada packages and encapsulation, software components, and multiple package bodies.

An approach being pursued is the generation of validation test programs from the Operational Definition. There is some question as to whether the CAIS Operational Definition can serve that function. Dr. Lindquist stated that a desire to use an automated tool omits the use of the natural language specification. The need for a formal semantic definition of the CAIS was discussed, but the ability to ascertain its correctness was questioned.

In characterizing his approach, Dr. Lindquist commented upon the concept of an automated assistant for generating validation test programs. The automated assistant would use Ada source code as input. The operational definition code feeds a procedure called IOGEN which goes through the source code and generates a set of input and output pairs. The input/output pairs and the source code are fed into a test program generator which generates validation test programs from these pairs. The resultant validation test programs have three different parts. The first part generates an initialization to do an initialization of the Node Model. The second one generates the test execution code. The third part then generates an analysis.

In summary, Dr. Lindquist stated that the CAIS Operational Definition should be comprised only of Ada, and the CAIS should be completed. Other needs to be investigated are the rehosting of tools, tools to be built on top of the CAIS, the functionality and efficiency of the CAIS, and the anticipated future environment needs.

The following points were made during the question and answer session that concluded the presentation.

- Ada is a good language to use in building tools.

- It would take a contractor a comparatively short period of time (graduate students took 2 months) to comprehend the materials used for the CAIS Operational Definition.

- The addition of another program to the CAIS OD such as another CVC test, to the O.D. involves some recompiling, but not of the entire system.

1.3 Presentation/Discussion of the CAIS Analysis Document

Gary McKee, Martin Marietta Aerospace
Standards Evaluation and Validation Working Group (SEVWG)

The presentation commenced with an introduction of the work undertaken by the Standards Evaluation and Validation Working Group (SEVWG). Mr. McKee, SEVWG chairman, then focused on the discussion of the CAIS Analysis Document. The CAIS Analysis Document provides the analysis, discussions, and concerns of SEVWG regarding MIL-STD-CAIS validation and evaluation. The document addresses areas to be considered and techniques to be employed in evaluating and validating CAIS Version 1, Version 2 expectations, and deferred issues.

The CAIS Analysis Document will undergo two major reviews. One review was completed during the E&V Team meeting in September and the second will be the KIT/KITIA Technical Interchange Working Group in December. The next version of the document is expected to be delivered in December of 1986. The contents of the CAIS Analysis Document can be partitioned into six areas which are context issues, validation, evaluation, evolution, and dependencies and testabilities. Mr. McKee's purpose in this discussion was to address the intent and content of the document, and the thought processes that went into the document's development.

In giving an overview of the contents of the document, Mr. McKee noted that the first four chapters of the document follow the boilerplate established for all E&V Team documents. These sections provide a brief description of the E&V Team, the SEVWG, and of the proposed MIL-STD-CAIS. The background, scope, and a description of the CAIS Analysis Document (CAD) are also provided. Contents of this document are intended to suggest methodology and provide a reference guide for conducting evaluations and validations of the MIL-STD-CAIS. It was noted that the section of the document that focuses on evaluation addresses the concept of evaluation metrics in terms of the kinds of things to be measured and constraints that are relevant to CAIS evaluation. CAIS evolution is addressed in a later part of the document.

Section Five of the CAD addresses validation issues, both technical and non-technical. The first concern addressed under technical issues deals with white box vs. black box testing. SEVWG maintains that the CAIS validation approach must not require access to source code, but should be able to define a set of tasks based on the semantics of the CAIS, and should be independent of the implementation. The non-technical issues section addresses such topics as policy concerns.

Constraints and Exceptions are two other topics addressed under "Issues" that require further definition.

It was noted that, whereas it is not required for the CAIS to be a full Ada implementation, Ada semantics should be adhered to in the interface which leads to exception handling.

Another issue addressed by the CAD is protocols. Three types of protocols (or interactions) exist within a CAIS implementation. These are protocols with the underlying operating system, protocols among CAIS interfaces, and protocols between APSE tools and the CAIS implementation.

A non-technical issue addressed involves how or whether to validate subsets and supersets. In discussing subsets and supersets, Mr. McKee noted that current AJPO policy prohibits use of subsets; therefore, validation of subsets will not be an issue. Supersets are generally defined as any combination of parameters that provide increased functionality. SEVWG's recommendation is that validation testing should not be required to detect supersets.

A major section of the CAD addresses MIL-STD-CAIS Validation. This section references the APSE Components Validation Procedures document produced by the SEVWG, and the Ada Compiler Validation Capability (ACVC) as guides to be used in conducting validations. This section of the CAD states SEVWG's understanding of the objectives of the CAIS Validation Capability (CVC) effort is to test conformance of CAIS implementations to the proposed MIL-STD, and to increase transportability of APSE tool sets.

The SEVWG recommends a three phase approach to validation testing; initialization using CAIS capabilities, testing of the interface, and examination of results. The test suite to be developed should be implementation independent. Mr. McKee noted that there are currently three significant implementations of the CAIS. They include implementations by Gould, MITRE, and the CAIS Operational Definition which is under the leadership of Dr. Lindquist of Arizona State University. These three implementations can provide information to the upcoming CVC effort.

The CAD addresses attributes that are subject to evaluation. These attributes will be derived from the E&V Requirements Document, the SEE Taxonomy, and the E&V Taxonomy document.

The section of the CAD addressing CAIS evolution identifies problem areas that exist in the current version of the proposed MIL-STD. Some of these 6 topics are:

- Distributed CAIS implementations
- Intertool interfaces and interfaces with the Run-Time Environment
- Security models
- Changes to the I/O section.

Mr. McKee noted that a major improvement to the current CAIS standard would be the addition of a set of well defined constraints to characterize the limits of an implementation.

Appendix D, "CAIS Dependencies and Testabilities", provides representative examples of the kinds of tests that should be considered in developing a validation test suite for the MIL-STD-CAIS. Development of this appendix provided SEVWG members the opportunity to acquire knowledge of the CAIS by performing detailed technical analysis of its various aspects. This study provided a Strawman approach to clarifying methodology issues. Mr. McKee concluded his presentation with a summary of subjects which SEVWG might study in the future. These include an examination of Descriptive Intermediate Attributed Notation of Ada (DIANA); a study of ARTEWG's work in the run-time environment; a look at other APSE tools, and the Graphics Kernel Support (GKS) interface.

1.4 The Performance Issues Working Group (PIWG)

Jon Squire
Westinghouse

Mr. Squire's presentation began with a brief outline on the activities of the Performance Issues Working Group (PIWG), and specifically their charter. He stressed the sensitivity of the decisions the group makes pertaining to the financial future of many companies. Mr. Squire stated PIWG's concern for professionalism, fairness, and an unbiased attitude, and yet to continue to present the facts. PIWG makes available benchmark tapes to everyone ensuring fairness.

In touching upon the results and problems in the technology, Mr. Squire stated that many procedures consist of over a hundred lines, the reason being the overhead of a procedure call needs to be high enough for the speed budget in the real-time application. This has led to big procedures.

Much heap space is being used and there is no attribute in MIL-STD-1815A designating space. This leads to a difficulty in measurement which then must be done indirectly. PIWG is taking different types of dynamic objects and accumulating data on the time for each.

In speaking of PIWG's methods, Mr. Squire stated that PIWG selects tests by a committee approach. Communication of future trends will be reported in the Ada letter. On the new tape to be distributed, PIWG has achieved automatic calibration.

Mr. Squire then proceeded to discuss PIWG's goals. PIWG is currently involved in measuring features and programming style comparisons. Their next subject is the measurements of applications which are executable programs using realistic data, and are useful for specific application. All data must be given along with the program in order for PIWG to run it.

Items in process are the Kalman filter, a communication package, and a package on navigation. There is also the possibility of an Ada math library which will probably be provided by the National Algorithms Group.

Mr. Squire perceives the real benefit of measurements as psychological. People will do a better job knowing their work is to be evaluated.

Of the two different styles of programming. Mr. Squire's view is that greater maintainability is achieved by controlling the number of flags in a given program.

Of the problems that PIWG faces, one issue is the credibility of the compiler benchmark. PIWG must know what is measured so that their statement on speed can be checked. Another problem is that what an author thinks is to be measured is not actually measured by the code. This is a fundamental problem of composite benchmarks. The application benchmarks are difficult to determine as the compiler writer and the benchmark author may not have the same view. With the whetstone it is important to do an internal procedure. Whether this is done or not makes a five to one difference in the timing of the benchmark.

PIWG has decided to not publish the speed given by manufacturers. Each manufacturer uses a different underlying method of rating, and, also, as there is a skew.

PIWG measures Central Processing Unit (CPU) time due to the variety of computers, multi-program computers, and multi-processing computers. Measuring CPU time is recommended for all machines that are not totally isolated. The tape PIWG has coming out will have CPU time for most computers, UNIX systems, VMS, Ada Language System (ALS), Rational 1000, and ALSYS. PIWG also measures ewall time. The PIWG maintains their results to be reasonably accurate. The scale Mr. Squire suggests for a safety margin is plus or minus ten percent. In response to a question, it was suggested that the Ada Compiler Evaluation Capability (ACEC) contractor should concentrate on using CPU time as opposed to wall time.

PIWG searched for a method to ensure an item could not be optimized out. A remote global has now been added in all benchmarks. There are four measurements made, the start of the control time, the end of the control

time, the start of the test, and the end of the test. The PIWG output has a standard boilerplate with three lines of descriptions, giving the output to the screen, to the disk, or to be saved in memory for embedding. PIWG has two duration IOs as none of the ICC front-ended compilers can substantiate a package at the library level. There is only one IO specification as all the code is the same, but there are three optional bodies. The code does not have to be totally transportable; it can be machine dependent.

The presentation concluded with a question and answer session. The following items were discussed at this time.

Mr. Squire was questioned as to the narrowness of the PIWG charter. He stated that this was deliberate as PIWG wanted to avoid subjective measurements.

In evaluations, PIWG considers the progress of the past two years in order to estimate the progress of the next two years.

The major technical obstacle standing in the way of a successful Ada compiler evaluation capability is nothing other than time, money, and talent.

With the close of this presentation, the E&V Team dispersed to their working groups.

2.0 THURSDAY, 4 SEPTEMBER 1986

2.1 Announcements

The Thursday session of the E&V Team convened with the following announcements by Ray Szymanski.

- Kathy Gilroy has resigned from the E&V Team.

- The December meeting will be mainly devoted to the CAIS.

- Once the contracts are awarded for the CVC and the ACEC, the focus of the team in this area will shift to an advisory capacity.

LCDR Myers commented on the new focus of the team, stressing the members role as advisors. He spoke on the decision-making process involved.

Ray announced that Lt. Robert Marmelstein will be the Engineer of Record Procurement for the ACEC and Jimmy Williamson will be the Engineer of Record for the CAIS Validation Capability.

Ray stated that on Friday LCDR Myers would speak, and then briefly opened the floor to questions. The E&V Team adjourned to their respective working groups for the remainder of Thursday's session.

3.0 FRIDAY, 5 SEPTEMBER 1986

3.1 TASC Update

Peter Clark
The Analytic Sciences Corporation

Mr. Clark opened his presentation by thanking those members who had submitted comments on the Classification Schema and requested comments from all team members. He followed with a review of the document schedule: the Classification Schema went out 15 August, the Reference Manual 31 August, and the Guidebook mailing planned for 15 September 1986. Temporary cutoff dates for team review were scheduled around upcoming E&V Team meetings: 15 November for the Classification Schema, 15 December for the Reference Manual, and 15 January 1987 for the Guidebook. Ray Szymanski should be advised if additional review time is needed.

A summary was given of the major changes to the Classification Schema and to the Reference Manual, derived from review commentary.

Changes to the Classification Schema included:

- Addition of background material originally contained in the Reference Manual which most members believe is more appropriate in the Classification Schema.

- Reordering material so the rationale is presented before the conceptual models for the Classification Schema and the Reference System.

- Defining requirements for the Reference System based on potential users of the documents and their perceived needs.

- Replacing the "conceptual model" figures with less confusing ones.

- Modifying several element definitions based on team input.

A table of contents was presented showing the source of the material used in the new schema (e.g., the E&V Workshop Report, previous drafts of the Reference Manual, etc.).

The six categories of perceived users of the Reference System were listed, along with a description of each. The six classes were 1) software acquisition personnel, 2) APSE/tool users, 3) APSE/tool builders, 4) E&V technology users, 5) E&V technology builders, and 6) investors. A suggestion was made to divide these into primary and secondary users with the document geared toward the primary users.

Presentation of the new figure illustrating the conceptual model of the E&V Reference System evoked several questions and comments. There was still confusion over what the various elements were meant to represent.

Bard Crawford explained that the ultimate user will not be using the schema, he will be using the Reference Manual--and that is the document that needs to be clear, easy to use, and able to stand on its own. The schema is basically for the team's use in viewing the organization and construction of the Reference Manual and Guidebook. The Reference Manual will contain an explanation of the document and procedures for its use.

Comments were made concerning the difficulty in designing a figure to accurately describe mappings and the relationships between them. A pictorial concept of the organization may not be the best approach.

Changes to the Reference Manual included:

- Moving background material and a section on the goals and users of the document to the Classification Schema.

- Replacing the usage scenarios with procedures for using the Reference System.

- Adding the rationale for the inclusion of whole APSE issues. For now, this chapter simply aids in identifying the issues.

- Placing the E&V categories in the procedures section in lieu of devoting a formal chapter to them.

- Indexing generic tool categories as a cross reference to functions.

- Identifying relationships between various attributes.

The new table of contents for the Reference Manual was shown listing the various sources for the material.

The next illustration was intended to show the relationships between tools, life-cycle phases, functions, and attributes as embodied within the Reference Manual. Again, there was some uncertainty of meaning expressed by team members.

The relationships were described as direct and indirect. Direct relationships are those that have been cross referenced to each other, such as:

- Life cycle phases with deliverables

- Life cycle phases with functions

- Attributes with other attributes

- Attributes with functions

- Functions with tools

The following items were listed as indirect relationships:

- Life cycle phases with tools

- Deliverables with functions

- Deliverables with tools

- Attributes with tools

The presentation concluded with a page from the Reference Manual illustrating the organization of the information, some of which was designed with a view toward the eventual automation of the system.

3.2 Working Group Reports

3.2.1 Standards Evaluation and Validation Working Group (SEVWG) Status Report

Gary McKee, the chair of SEVWG, presented the following status report for the group. The current members are Gary McKee, Mike Mills, Tim Lindquist, Jeff Facemire, and John Reddan; Kathleen Gilroy has resigned. The group was visited by Suzanne Menichiello. There were no deliverables due this quarter. The deliverable for next quarter is the review of the CAIS document at the December meeting and its delivery to Ray Szymanski. Action items for the group were noted.

3.2.2 Requirements Working Group (REQWG) Status Report

The REQWG vice-chair, Marlene Hazle, presented the status report for the group. REQWG has a new member in Captain Pickart who is replacing Richard Faris. There were three visitors to the group, Fred Francl, Suzanne Menichiello, and Jon Squire. There were no deliverables due this quarter; the deliverable due next quarter is Version 1.0 of the Tools and Aids Document. The projected work includes revising and commenting on the Tools and Aids Document, and considering mechanism for the continuing assessment of needs. There are no presentations planned. REQWG accomplishments include the delivery of Version 2.0 of the Requirements Document, a new draft of the Tools and Aids Document, several scenarios depicting the use of or need for E&V technology, a report on Software Engineering Institute (SEI) evaluations, a report on the internal evaluation effort, and a review of the evaluation attributes in the Requirements Document and the Classification Schema. Key issues announced by REQWG were the Tools and Aids Document, a description of the team and task role of E&V technology, possible approaches to fulfilling the information dissemination function, and the coverage of the compiler evaluation requirements by the ACEC contractor. Individual action items were also noted.

3.2.3 Coordination Working Group (COORDWG) Status Report

The COORDWG status report was given by Don Jennings, chairperson. The deliverable for this quarter was the Technical Coordination Strategy Document, Version 3.0. The accomplishments for the quarter were the public coordination strategy document, Version 3.0 delivered to Ray Szymanski, the E&V minutes and the status report, and the E&V public report, Volume II. Concerning unresolved problems, COORDWG remained unsure of the status of the vice-chair. The projected work includes the Technical Coordination Strategy Document, the new status reports, and the minutes. There were no deliverables due other than the standard minutes. The key issues addressed this quarter were the Technology Coordination Strategy Document, Version 3.0, and the new format the status reports. The individual action items were noted. There were no presentations planned for next quarter and no other significant information.

3.2.4 APSE Working Group (APSEWG) Status Report

The APSEWG accomplishment for the quarter was their work on the APSE Information Gathering Plan and a letter to vendors. There were ro deliverables. The key issues addressed were the "whole APSE" issues in the Reference Manual, and the APSE Information Gathering Plan and the letter to the vendors. There are no deliverables due next quarter. The projected work for next quarter is the finalization of the plan and the vendor letter. There were no unresolved problems and individual action items were noted..blank 2

3.3 AJPO Update

LCDR Philip Myers

LCDR Philip Myers gave the following update on the activities of AJPO. LCDR Myers stated that there has been significant progress on Ada in the Department of Defense due to some recent developments. LCDR Myers was hopeful that the Navy policy would soon be signed. He stated that along with this policy will be the adoption of DOD-STD-2167 as the formal documentation standard; Ada would become the single high level language used within the Navy.

LCDR Myers related the following information concerning current and upcoming events. He commented that the validation criteria is under review by the Evaluation and Validation panel of the Ada Board. In a continuing series of meetings at NATO, Ginny Castor and Colonel Taylor were again meeting on finalizing the terms of reference in the Memorandum Book Agreement between the nations participating in the work on CAIS and the evaluation technologies. In November, an Ada exposition will be held in Charleston, West Virginia. Secretary Weinberger is scheduled to speak; Senator Byrd is a co-sponsor of the event.

LCDR Myers explained that there is a problem with the APSE in the functional taxonomy in the APSEWG document due to a previous restriction. He stated that AJPO is trying to get the restriction reversed. If the SEE taxonomy is reversed there will be no restriction on the APSE.

The floor was then opened to questions from the team. He stated in response to a question that there is no reason not to use the Ada networks. He also explained that the Ada marketplace needs verification and formalization of technology.

LCDR Myers closed the discussion stressing the importance of the CAIS work stating that it will become a more significant item.

3.4 Closing

In the absence of Ray Szymanski, LCDR Myers thanked everyone for attending. The E&V Team meeting for September 1986 was adjourned.

APPENDIX A

ACTION ITEMS AND RESOLUTIONS FROM THE JUNE E&V MEETING

AI-06-06-86-1	Systran. Mail presentation material to attendees.	Accomplished
AI-06-06-86-2	Systran. Put draft of June minutes on the NET.	Accomplished
AI-06-06-86-3	Systran. Prepare minutes of the REQWG and SEVWG meetings.	Accomplished
AI-06-06-86-4	Szymanski. Contact ARTEWG about July meeting.	Accomplished

APPENDIX B

SEPTEMBER ACTION ITEM LIST

AI-09-05-86-1	Systran. Prepare and distribute minutes for the REQWG, SEVWG, and general session.

APPENDIX C

ACRONYMS

ACEC	Ada Compiler Evaluation Capability
AJPO	Ada Joint Program Office
ALS	Ada Language System
APSE	Ada Programming Support Environment
APSEWG	APSE Working Group
ARTEWG	Ada Run Time Environment Working Group
CAIS	Common APSE Interface Set
CLI	Command Language Interpreter
COORDWG	Coordination Working Group
CPU	Central Processing Unit
DIANA	Descriptive Intermediate Attributed Notation of Ada
E&V	Evaluation and Validation Team
GKS	Graphics Kernal Support
KAPSE	Kernal Ada Program Support Environment
KIT/KITIA	KAPSE Interface Team/KAPSE Interface Team Industry & Academia
OD	Operation Definition
PIWG	The Performance Issues Working Group
REQWG	Requirements Working Group
SEI	Software Engineering Institute
SEVWG	Standards Evaluation and Validation Working Group

APPENDIX E

DOCUMENTS DISTRIBUTED AT MEETING

1. Agenda for 3-5 September 1986 E&V Team meeting.
2. E&V Attendance/Membership List.
3. E&V Status Report.
4. Minutes of the June E&V meeting.
5. Informational material concerning location of the San Diego meeting.
6. Presentation materials used in "CAIS Operational Definition".
7. Presentation materials used in "MIL-STD-CAIS Analysis Document".
8. Presentation materials used in "Performance Issues Working Group Status Report".
9. Presentation materials used in "E&V Technical Support Activities: Report To E&V Team Meeting".

APPENDIX E

LIST OF ATTENDEES

Brookshire, Jerry R.
TI
P.O. Box 660246, W/S 3114
Dallas, TX 75266

Clark, Peter
TASC
1 Jacob Way
Reading, MA 01867

Crawford, Bard
TASC
1 Jacob Way
Reading, MA 01867

Francl, Fred
Sonicraft
8859 S. Greenwood
Chicago, IL 60619

Gicca, Greg A.
GTE/WIS
1 Federal St.
Billerica, MA

Harto, Debra L.
AFATL/FXG
Eglin AFB, FL 32542

Hazle, Marlene
MITRE Co.
Burlington Rd.
Bedford, MA 01730

Jennings, Don
OC-ALC/MMECO
Tinker AFB, OK 73145

Kean, Liz
RADC/COEE
Griffiss AFB, NY 13441

Kirkbride, Kathy
Systran Corporation
4126 Linden Ave.
Dayton, OH 45432

Leavitt, Thomas C.
Boeing Military Airplane Co.
P.O. Box 7730 MS K75-67
Wichita, KS

Lindquist, Tim
Arizona State University
Dept. of Computer Science
Tempe, AZ 85287

Maher, Patrick J.
Magnavox Elect. Sys.
TC10-C10D542
1010 Production Rd.
Ft. Wayne, IN 46808

Martin, Ronnie
Georgia Tech
Software Eng. Research Ctr.
Atlanta GA 30324-0280

McKee, Gary
Martin Marietta
Info & Communication
M/S L1640 P.O. Box 179
Denver, CO 80201

Menichiello, Suzanne Prudhomme
The Aerospace Corp.
2350 El Seguende Blvd.
El Segundo, CA 90245

Myers, LCDR Philip
AJPO
3D139(1211 Fern St), Pentagon
Washington DC

Pickart, Bruce R.
HQ AF OPR TEST CTR
Kirtland AFB, NM (LG5)
Albuquerque, NM 87117

Reddan, John
SYSCON Corp.
8990 Sherman St.
San Diego, CA 92110

Shirley, Jane
Systran Corporation
4126 Linden Ave.
Dayton, OH 45432

Tompkins, Mary A.
Lockheed
2124 East St Elmo Rd.
Austin, TX 78744

Wills, Betty
CCSO/XPTB
Tinker AFB, OK 73145

Rhoads, Barbara L.
Systran Corporation
4126 Linden Ave.
Dayton, OH 45432

Squire, Jon
Westinghouse
P.O. Box 746 MS 1615
Baltimore, MD 21203

Weiderman, Nelson
Software En.. Inst.
Carnegie Mellon University
Pittsburgh, PA 15213

APPENDIX I

MINUTES

of the

EVALUATION AND VALIDATION (E&V) TEAM MEETING

3-5 DECEMBER 1986

The task for the Evaluation & Validation of Ada* Programming Support Environments (APSEs) is sponsored by the Ada Joint Program Office (AJPO).

*Ada is a registered trademark of the U.S. Government (Ada Joint Program Office)

TABLE OF CONTENTS

1.0 WEDNESDAY, 3 DECEMBER 1986 . . . . I-3
    1.1 Welcome, Introductions and General Business I-3
    1.2 CAIS Presentation I-3
    1.3 Experiences With CAIS I-8

2.0 THURSDAY, 4 DECEMBER 1986 . . . . I-10
    2.1 E&V Classification Schema I-10

3.0 FRIDAY, 5 DECEMBER 1986 . . . . I-11
    3.1 AJPO Update I-11
    3.2 Working Group Reports I-13
        3.2.1 Coordination Working Group (COORDWG) Status Report I-13
        3.2.2 Ada Programming Support Environment Working Group (APSEWG) Status Report I-13
        3.2.3 Requirements Working Group (REQWG) Status Report I-13
        3.2.4 Standards Evaluation and Validation Working Group (SEVWG) Status Report I-13
    3.3 Open Discussion I-14

APPENDIX A Action Item List from the December E&V Meeting . . . I-15

APPENDIX B Speech by Caspar Weinberger I-16

APPENDIX C List of Attendees I-21

1.0 WEDNESDAY, 3 DECEMBER 1986

1.1 Welcome, Introduction and General Business

The Evaluation and Validation (E&V) Team meeting opened with the introduction of Jim Parlier (General Dynamics) by Ray Szymanski. Mr. Parlier welcomed the team to the General Dynamics facilities, then oriented the visitors to the area. After the orientation he gave a brief presentation on the Data Systems Division of General Dynamics.

Ray extensively restructured the agenda, then introduced a key issue of the meeting: reorganization of the E&V team. The team's past efforts have been directional. However, a reorganization along product lines was under consideration. Ray announced plans to talk with each working group for their input. A request for suggestions resulted in the comment that the focus should be on technology transfer, which would include:

- The Guide Book, Reference Manual, and Tools and Aids document;

- A mini-course covering the application of E&V technology to accompany RFPs.

Ray stated that the Ada Compiler Evaluation Capability (ACEC) contract was due to be awarded around 15 December, and contract award for the CAIS Validation Capability (CVC) is expected around 1 January.

Guests and visitors introduced themselves. Visitors included Tricia Oberndorf and Hans Mumm from the Naval Ocean Systems Center (NOSC); Shawn Fanning, Gary Pritchett and Geoff Clow from Softech. Ray introduced John Stanton of the Ada Joint Program Office (AJPO).

1.2 CAIS Presentation

Patricia Oberndorf
NOSC

Patricia opened her presentation with a brief review of terms. CAIS 1 and MIL-STD-CAIS are used to refer to DoD-STD-1838. CAIS 2, CAIS A and 1838-A are commonly used to refer to SofTech efforts.

Around 1981-1982, the AJPO had the Army (ALS) and the Air Force (AIE) working on Stoneman Ada Programming Support Environments (APSEs). These APSEs different and required extensive rewrites to accomplish tool transportability. This was deemed inefficient and a new plan was developed. The goal of this new effort was to design tools along a set of common interfaces (either ALS or AIE). Then, by creating a set of common interfaces, transportability and reliability could be maximized. This would also allow ALS and AIE to evolve.

The Navy was selected to accomplish this task for several reasons. They didn't have a Stoneman environment, but they had expertise in environment development and five year's research in the area.

Two teams were formed: the KAPSE Interface Team (KIT) and the KAPSE Interface Team from Industry and Academia (KITIA). The KIT is comprised of about 30 government and paid contract personnel. Six working groups were formed: Stoneman (STONEWG); Requirements and Design Criteria (RACWG); Compliance (COMPWG); Guidelines and Conventions (GACWG); Definitions (DEFWG); and the Common APSE Interface Set working group (CAISWG). The CAISWG's aim was to develop a MIL-STD based on the CAIS Rationale (RAC), and had a subgroup called the CAIS Editorial Board (CEB). The KITIA has served its purpose and has been officially disbanded, although several people are still actively involved on an informal basis.

The chronology of CAIS development is as follows:

DEC. 81: OSD/Tri-Service Memorandum Of Understanding

JAN. 82: First KAPSE Interface Team (KIT) meeting

FEB. 82: First KAPSE Interface Team from Industry and Academia (KITIA) meeting

SEP. 83: CAIS 1.0 Public Review
CAIS 1.1
(= 1.0 + correction of errata)

JUN. 84: CAIS 1.2 (predominantly European distribution)

AUG. 84: CAIS 1.3 Public Review

NOV. 84: CAIS 1.4 Public Review

JAN. 85: Proposed MIL-STD-CAIS.

CAIS is a set of interface specifications (not program code, but a document). This document will be used by tool writers to develop host-independent tools. It is open-ended in that more interfaces to the I/O section may be added. Currently it is limited to three terminal types and mag tape. CAIS is also limited to operating system-like services that are amenable to host-independent agreement. Documentation will include a Reader's Guide and Rationale.

Many CAIS projects are underway. The government-sponsored projects include: the Operational Definition at Arizona State University; TRW; MITRE; SofTech; the VHSIC Hardware Description Language (VHDL); Andyne and the University College of Wales (UK mod). MITRE has two prototypes. The MITRE McClean version is a 75% implementation without process control and all the I/Os. Mitre Bedford's version is a non-piggyback CAIS. It is a bare-machine implementation on an IBM PC with an 80386 microprocessor. Independent efforts include Gould, Intermetrics, Honeywell, IBM, TI and the University of Houston/Clear Lake.

CAIS is stated in ADA package specifications. It is not a run-time environment (RTE) or an OS but it has features of both. The main goal of CAIS is to improve transportability; as a result, a CAIS OS would be infeasible. As currently implemented in piggyback fashion, CAIS is an interface to OS-level services. Installation on a bare machine would require adding features to fulfill OS functions, thus hampering transportability.

DoD-STD-CAIS document organization is as follows:

1. Scope;
2. References;
3. Definitions;
4. General Requirements;
5. Detailed Requirements;
   5.1 General Node Management,
   5.2 CAIS Process Nodes,
   5.3 CAIS Input and Output,
   5.4 CAIS utilities;
6. Notes

APPENDICES.

The CAIS document embodies the node model concept, and provides support in three main areas. They are: Administration of entities (5.1); Process Control (5.2); and I/O and Device Control (5.3).

Several major concepts and themes underlie the development of CAIS. One key concept is host OS independence. CAIS must be implementable on any host, either a bare machine or piggybacked on the host OS. It must be suitable to Ada tool writers and allow smooth feature integration. The 90/10 rule must be applied for coverage of tool/OS services. This rule states that 90% of the interfaces must be available for 90% of the tools 90% of the time. CAIS must provide a framework for future extensions and implementation freedom, as well as a flexible configuration management foundation. It has to merge modern OS and DBMS ideas using a simple, uniform underlying concept (the node model). Finally, its interface level must be high enough to allow piggyback implementations for efficiency, yet low enough for bare machine implementation. This allows effective tool management.

A recently completed survey solicited comments on the CAIS document. The responses were categorized into twelve areas:

1. Editorial (missed punctuation, etc.);
2. Explanations of concepts/ terms;
3. Naming conventions;
4. MIL-STD-962A conformance;
5. Procedural comments;
6. CAIS Package Structure;
7. Exceptions granularity (name errors, etc.)
8. Pragmatics expansion;
9. Interface enhancements;
10. CAIS Access Control;
11. CAIS List Management, and
12. I/O reorganization.

The first four items were determined to have no technical impact. Items five and nine imply changes to be made. The other points are significant, requiring changes to intent and meaning, thus improving the CAIS approach.

Under item three (3), the naming/parameter order, these changes were suggested:

- Generation of conventions for naming;
- Generations of conventions for ordering parameters;
- Change existing names according to conventions;
- CAIS study note on naming conventions;
- Consistency;
- Descriptive package names;
- Type names: ... _Type;
- Ada object names: nouns/noun phrases;
- Procedure names: verbs;

- Descriptive attribute and relation names, and

- Default parameters listed last.

The rationale is that names in CAIS (as with Ada keywords) will be important, and that semantic identification in context enhances reliability.

Proposed disposition applicability (item 4C) is in four parts. First, a new subsection should be included in section 1 called Applicability. Second, proper applications would include prototype implementations of CAIS and tools, implementations/comparison studies, and experimental usage studies. Third, CAIS is not intended for use by any project whose main application is not CAIS experimentation. Finally, CAIS is not for use in deployed systems.

Item six (6), on CAIS Package Structure, has pointed out a problem with the MIL-STD-CAIS nested package. The implementation's structure is called monolithic, limiting user visibility of individual CAIS packages and control of them. As a result three changes are proposed. First, remove the outer-package CAIS. Second, leave the remaining packages in a parallel structure and finally, prepend all package names with CAIS.

The CAIS Package Structure rationale has several important points. The current monolithic structure and its resulting severe run-time penalty must be reduced. Tools will need improved run-time efficiency as well. Finally, a minor problem arose concerning package structure. This problem had considerations based in Ada and concerned the use of limited private types in each package. Packages need insight into other package's types as they are no longer one large structure.

Pragmatics limits (item 8) in the proposed MIL-STD-CAIS are insufficiently specified. The list of CAIS implementation property limits must be expanded, and a Pragmatics package should be added. CAIS was defined in terms of the smallest upper bound allowed. Implying that no more than ten relationships could emanate from a node meant that any implementation could have more than ten but no less. Now there are two kinds of constants. The CAIS-defined limit defines the smallest upper bound allowed, while the implementation-defined limit is the actual upper bound supported. The CAIS-defined limit must be less than or equal to the implementation-defined limit. Also, two new exceptions for implementation-defined limits are supported.

The Pragmatics rationale has several points. It allows an increased set of implementation properties, easing implementation into the specification. Consistent naming of constants and exceptions is achieved. A single, unrealistic limit is avoided and improved information allows better tool transportability. Finally, these points allow indication of implementation violations.

Item eleven (11), List Management, had the following subissues: list copying inefficiencies, storage management, awkward interfaces and imprecise value semantics. Serious problems arise in MIL-STD-CAIS operations, as these require extensive list copying and reconstruction to modify nested items in nested lists. As these are high-volume actions, tool performance would be dramatically impacted. The solution is to clarify the notions of a linear list and the nested sublist/nested list structure. A new concept was added, called the current linear list. This allows the pointer to move to a sublist, allowing free manipulation at that level.

Tricia concluded by a problem with storage management. The list management interfaces wouldn't allow reclamation of implied storage used for lists when the storage was no longer needed. Consequently, a new interface procedure was adopted that allows CAIS to recognize free space and de-allocate it.

1.3 Experiences with CAIS

Hans Mumm
Naval Ocean Systems Center

The discussion centered around three MIL-STD-CAIS prototypes: MITRE, These implementations have been received by NOSC; however, only two are up and running.

The first one received was the MITRE July 86 version. It has no mandatory/discretionary access control or process control, but allows multiple users. This version was used on VAX with the ULTRIX OS, utilizing a Verdix data compiler. It was converted to run on a VAX/VMS using a Digital Equipment Corporation (DEC) Ada compiler.

Installation of the MITRE prototype took forty-one (41) steps. This was necessary because tests had to be performed as certain parts were installed. The original documentation was sketchy; new documentation was written during installation by a student employee. An abbreviated list of the steps required for installation follows:

1) Get tape and installation instructions;

2) Copy CAIS and test files to disk;

3) Compile CAIS and C files;

4) Link and run NEW_USER;

5) Compile, link and run test programs, and

6) Link and run CAIS commands.

Two tools were converted to run on the MITRE prototype: the Ada line editor and Unpage, which unconcatenates large Ada files. The Ada line

editor was written by TI but was converted by Chuck Howell at MITRE. These tools had file handle. Also, both programs were written with positional notation; this had to be swapped with CAIS positions. The biggest problem with conversion was that some features weren't implemented by MITRE. This caused some unpleasant surprises during experimentation.

The October 86 version of the CAIS OD was discussed. It doesn't include I/O interfaces and only provides single-user operation. It is VAX/VMS based using a DEC Ada compiler. Tool implementation is possible two ways: as an Ada task or as a separate program.

Installation of the CAIS OD was more efficient due to excellent documentation. It took two days to get the OD up and running using the following steps:

1) Get tape of CAIS OD and command files with instructions;

2) Copy files to CAIS OD directory;

3) Compile and link CAIS files.

There were two methods available for converting Ada tools. The Ada task method was chosen as it required fewer steps. Programs tested on the OD included the Ada line counter and classroom programs written by students. In addition, some Ada tools for VAX UNIX were used after rewriting them from scratch.

The Gould prototype is a nearly whole implementation except for mandatory/discretionary access control, form terminal and mag tape. It runs on a SEL computer and has two versions, MPX and UTX. Originally the Gould prototype used the unvalidated Irvine Ada compiler, although Gould now has a validated compiler. It is 97% in Ada, with process control in C.

Installation doesn't appear to be a problem as the documentation seems fairly complete. To obtain a prototype, the user pays $500 for a tape, license and documentation. The tape includes source code, an object file and demos, such as the Tower of Hanoi. To implement on VAX/VMS, one needs a UNIX OS with an Ada compiler to utilize this prototype. All implementations will be upgraded to comply with DoD-STD-1838.

At the conclusion of Mr. Mumm's presentation, the E&V team dismissed to individual working groups.

2.0 THURSDAY, 4 DECEMBER 1986

2.1 E&V Classification Schema

Dr. Bard Crawford
The Analytic Science Corporation

Bard opened the general discussion of the Classification Schema by focusing on the figures. Four suggestions were made concerning overall use of the figures:

1) Use the figures within the document as is currently done;

2) Use the figures separately as a Reader's Guide to the User's Manual or Guidebook Reference;

3) Use the figures as an introductory chapter in the Reference Manual,

4) Use figures both as a Reader's Guide and an introductory chapter.

Comments about the individual figures were solicited. Some felt figure one (1) was an attempt to define non-existant things; that the term indexes is not sufficient. Also, omission of the on-line system concept was called for.

Figure three (3) was deemed too abstract. Examples were called for, such as the library card catalog concept as was done in an earlier version. The term Formal Chapters in figure five (5) drew several negative comments. Most thought this term should be changed to Reference Material. The taxonomies in figure six (6) should be compared to a card catalog with the differences illustrated. The term index needs further explanation, as in figure one (1).

Figure seven (7) generated the most discussion. For example, a user may only want to know about compilation speed, not the whole process. Simple References should be renamed to Other Relative References, giving a better idea of what is in the text frames. Also, there was a question about what Guidebook References are. This should be compounded and related to a different Guidebook section.

In figure eight (8), the second line should be dropped. The computer-based, interactive version is irrelevant until it becomes available. The figure itself has no in-arrows and the intersections (I,J) and (J,K) should be labeled. Figure nine (9) was deemed too abstract.

The phrase evaluation metric in figure ten (10) should be changed to subject metric. Figure eleven (11) text seemed to have too many terms (e.g. function index, taxonomy) that need explanation.

Two other general comments were made. First, the schema has too much terminology. Second, a background of development is unnecessary and should be placed in an appendix.

3.0 FRIDAY, 5 DECEMBER 1986

3.1 AJPO Update

LCDR Philip Myers
Ada Joint Program Office

LCDR Myers opened by thanking General Dynamics and praising their Ada involvement. Then he introduced four new members to the AJPO: John Stanton, Sebastian Ramono, Barbara Flemming and Ray McClendon.

John Stanton came to the AJPO from the Federal Software Testing Center, whose function is being transferred to the National Bureau of Standards. He was involved in Ada, Pascal, FORTRAN and COBOL as well as validation and test suite development. He is designated to assist Philip in the task area of monitorship/sponsorship as Philip is leaving in July. John will be a part of E&V, adding to the AJPO's base of experience.

Sebastian Ramono (Bardie) will be assisting in the Ada Validation Office. He came from the Naval Ada Air Systems Command, and was involved in embedded computer resources management in avionics computer resources. So far, he has assisted John in wrapping up the Ada Compiler Validation Procedures and Guidelines, which should be released 12 December. This document introduces several concepts, including derived validations and project-validated compilers.

Barbara Flemming came over to AJPO from Defense Communications HQ. She was involved in support of WIS and Ada, and will assist in the KIT and CAIS areas. Also, Barbara will assist in international activities, namely the International Standards Organization (ISO) Working Group 9.

Ray McClendon is the newest member of the AJPO. Formerly with the Army CECOM in Fort Monmouth, he will replace Tom Sheehan. His areas of responsibility will include the ARTEWG and budget.

In general business, Philip announced that the NET host will move from DEC-20 to UNIX. This change is forthcoming because of cost considerations. The implementation of host nodes on USNET, CSNET, MCI Mail and other systems will be attempted. The AJPO will still be a DDN host.

Working Group 9 of the ISO is a European group similar to ANSI. It has been working to make Ada an international standard. In a related development, the Language Reference Manual has been translated into French, Japanese and German.

The Ada Rationale has been received and is currently under review by the Ada board. Announcement of its availability will be made via the <Ada-INFO> directory on Ada 20, the Ada-IC newsletter and other sources.

The Ada board's charter is in the renewal process. After the February meeting a new board will be seated. As interpretations come down and are approved by the Director of AJPO, they will be incorporated into the Validation Suite. The board has recommended that run time and evaluation issues be given priority.

Philip defined the AJPO's role as primarily a technical staff which has limited authority over policy matters. The AJPO will staff certain policy documents for higher authority within DoD to approve. This is due primarily to a limited annual budget. Despite this, they do have several efforts underway Team (ASEET).

This team is developing curricula to get Ada into academies and schools. One product of the team's efforts is a video course. The AJPO provides no funding to the Armed Forces for personnel education in Ada use. They manage the infrastructure to support the language, but are not developers. They are to be considered an informational clearinghouse. Once the language is standardized, it will be turned over to industry. Overall, the AJPO's aim is to organize work to minimize duplication of effort, freeing the Ada Board to run the task forward.

Run-time has been a major concern to the Ada community. Due to poor performance, no vendors can or seem willing to provide run-time specifications. The E&V Team and the ARTEWG have helped bring attention to this area, and improvement is expected. Also, the ARTEWG has been working on a white paper at the AJPO's request. This is basically a document that identifies problem areas that need funding help. The next version of this document will surface at the February Ada Board meeting.

Secretary of Defense Caspar Weinberger understands the importance of the program and supports policy concerning Ada use. Currently all services have directives on Ada use. General Salsbury from the Army says Ada is the only language to be used, even for information systems. The Air Force has released their 800-14 regulations, and the Navy requires Ada for new starts and major upgrades. Navy off-the-shelf systems must use Ada. Although these policies are out, because they are new, many contracting officers aren't aware of them. Once reviewed, these policies will assist in the introduction of Ada. The Department of Defense policy concerning Ada is planned for release by the end of the year.

The NATO development effort is going well. The U.S. has provided a courier to hand carry the Ada MOU. Ten other nations are working with the U.S. on an APSE based on DoD-STD-1838. It is a very rich subset that includes single-level security. Colonel Taylor is the U.S. representative for this effort, and Virginia Castor is the chairperson. A management team comprised of prototypers and personnel from MITRE, IDA, and AJPO are preparing to meet next month. The NATO group also plans to develop a NATO interface standard and the requirements for it.

Some confusion has arisen concerning the types of information to be shared with other countries. The NATO Standard is to be widely distributed; however, items designated controlled technology are distributed according to a formal agreement established with that government. Products of the NATO effort will be varied and will include tool sets, equipment, evaluation technology and an operational scenario development. Only NATO countries that signed up for this effort can access the software products that result from it.

3.2 Working Group Reports

3.2.1 Coordination Working Group (COORDWG) Status Report

Pat Maher from Magnavox gave the COORDWG report. Their deliverable due was the Technical Coordination Strategy Document. Activities reported were a meeting with APSEWG to discuss the proposed reorganization and conducting a survey. Also, a letter drafted by APSEWG was reviewed. This letter solicits guest APSE tool vendors to speak at E&V team meetings.

3.2.2 Ada Programming Support Environments Working Group (APSEWG) Status Report

Peter Clark of TASC delivered the APSEWG report. His group drafted the letter reviewed by the COORDWG to solicit guest speakers. The APSEWG discussed reorganization plans as well as ACEC issues.

3.2.3 Requirements Working Group (REQWG) Status Report

The REQWG report was given by Helen Romanowski of Rockwell. Her group drafted a general statement of the E&V task, differentiating between team and task. The hierarchy of attributes used to review APSE components was discussed. A report exploring the government survey process was given as well. There were two main issues. The Tools and Aids document survey was reviewed, and the need for assessment mechanisms was discussed. Also, since certain parts of the E&V document are now defined as controlled technology, the next E&V report may have limited distribution.

3.2.4 Standards Evaluation and Validation Working Group (SEVWG) Status Report

In the SEVWG report, Mike Mills stated that CAIS validation issues generated intense discussion. SofTech gave a presentation on planned development of CAIS A. A key element of the meeting was distinguishing between validation and evaluation. Also, the CAIS Analysis Document was reviewed with the following conclusions:

1) Extensive revisions are necessary, including a title change to: "Issues and Strategy in CAIS E&V";

2) The CAIS-REV-A (DoD-STD-1838A) may be complex to validate;

3) The Working Group plan is to extensively revise the present CAD documents.

3.3 Open Discussion

During the open discussion period, two items of interest were identified. Because all team members are very interested in the two upcoming contractual efforts, announcements of contract awards will be made over the NET. Then, as several team members were not familiar with AJPO's policy on the control of technical information, the policy was identified and explained.

The meeting was then adjourned.

APPENDIX A

ACTION ITEM LIST
FROM THE
DECEMBER E&V MEETING

AI-12-86-1 R. Szymanski. Send copies of the document from the United Kingdom to all government members of the E&V team and to MITRE.

APPENDIX B

SPEECH BY CASPAR WEINBERGER, SECRETARY OF DEFENSE
at the
ADA EXPO 1986 Conference, 19 November 1986

Good morning, ladies and gentlemen, it is a pleasure to join you today for this conference and to speak to you, the representatives of the Ada community, on the importance of the Ada computer language to our nation. It is also a pleasure and honor to speak in the home state of Senator Robert Byrd, who has been such a distinguished representative of this region for so many years.

Forty-six years ago, the United States was asleep under the delusion of isolation. However, President Franklin Roosevelt and men from the scientific community, like Vannevar Bush and President James B. Conant of Harvard, recognized that if the United States and its allies were to prevail against the forces of tyranny, our nation and our allies would have to harness our scientific and engineering skills, our industrial potential, and our military organization in a desperate race against time. What resulted was an unprecedented partnership among scientists, engineers, industrialists and military officers. From this partnership came tremendous innovations in weapons systems, and, yes, even computer systems, which contributed so substantially to victory.

Winston Churchill's splendid words after victory in the Battle of Britain applied as equally and justly to British scientists as they did to the Royal Air Force. If never in history had so many owed so much to so few, they owed it not only to the magnificent skill, encourage and endurance of British aviators, but also to the small group of scientists who had developed the radar warning system that gave Britain's outnumbered fighter pilots the edge they needed.

With this history of cooperation between U.S. science and the defense establishment as a precedent, Senator Byrd's leadership in the of a Software Valley here in West Virginia comes as no surprise. Senator Byrd recognizes that our national security posture is more than the aggregation of our ships, tanks, aircraft and service personnel. Rather, true national security rests not only on our deployed military strength, which Senator Byrd has helped us rebuild, but also on our political will as a nation, our underlying economic strength, and our scientific and technological creativity. By bringing together, in an enterpreneurial spirit, participants from academia, the computer hardware and software industry, and government, Software Valley is as great an investment in our national security as an aircraft manufacturing plant or a shipyard. It is in Software Valley that we have an identical cooperative to that which saw us through World War II.

Today we are in the midst of a technological revolution of major proportions. Microelectronics, biotechnology and the information explosion already have had a profound influence on our national security posture for the rest of this century and well into the next. Yet the

technological innovations upon which defense now relies were not precipitated, nor achieved, by government alone.

For this reason, it is gratifying to see representatives from throughout the Ada community-- DoD, industry and academia-- here today. Your presence is indicative of the necessary widespread support for the Defense Department's effort to make Ada the standard, high-order computer programming language for use in defense systems. This is especially important because the computer has become the principal means for enhancing productivity in and out of government-- and enabling the use of innovative technology. I am told that the computer industry has achieved a hundredfold increase in effectiveness coupled with more than a tenfold decrease in cost during the past decade. In fact someone calculated that if automobile manufacturing costs had kept pace with computer costs, a car that cost $3500 in 1953, would cost $3.50 today. Computers have been incorporated not only into the operations of industry, and into the offices where the work is planned and managed, but also into the products we see in the marketplace.

Computers have affected all our lives. Nowhere is this more apparent than in defense. In a complex world as instantaneously dangerous as it is today, where the demands for knowledge are so great, we appear to be in a race between demand for information and the ability of our information to command and control systems to provide it.

Computer technology has had a profound impact on the Defense Department. The importance of the computer to those technologies that development of high capability weapons systems cannot be overestimated. Perhaps not as visible as weapons systems, but just as vital, is the management and logistics side of defense-- and computer technology has an important role here, too. Through such initiatives as the Computer-Aided Logistics Support programs (CALS), we expect to see in the 1990's major gains in efficiency, and reductions in the mountains of paper needed to procure and support our weapons systems.

Our interest in computers is not new. The Defense Department had a very direct involvement in the early development of the digital computer-- indeed the world's first all electronic digital computer-- ENIAC-- was developed for the sole use of the Defense Department. Since the 1950's DoD has had a vital interest in computer technology development for several purposes:

- Scientific applications, where the intensive computational needs of weapons system design have served both as a driver of, and a major market for, scientific computers.

- Weapon system applications, where embedded computers are part of virtually every new weapon system. These embedded computers give us a clear, qualitative edge over our adversaries.

- Command, control, communications and intelligence (C3I) applications. Modern digital communications are the central nervous system of our military capabilities.

- General purpose applications, where we use computers in many of the same ranges of business, industrial, and other activities that the private sector does.

- Finally, for the future, we look to a vastly increased computer capability to enable us to make the President's Strategic Defense Initiative work and to give the world the most hopeful strategic concept in at least the last forty-five years.

Time does not permit me to go fully into all of our programs in these application areas. Suffice it to say that DoD is continuing its role of national leadership with its sponsorship of such programs as Very High Speed Integrated Circuits (VHSIC) technology, DARPA's strategic computing program, and Ada, DoD's common high-order language for our weapon's systems of the 1990's.

As you know, more than a decade ago, the Defense Department faced a number of significant challenges with its computer systems. Computers had proliferated throughout the services and the unified and specified commands. Each computer system required unique software, but documentation was poor or non-existant. Furthermore, we had almost no ability to modify programs to allow interaction between machines or systems. As a result, vast numbers of hybrid computer systems were in use that were very costly and were very inefficient. The need for a common computer language was obvious.

You are all aware of the history of the development of Ada and the work by the Ada designers and distinguished reviewers-- many of whom are here today. We are grateful for their contributions to the development of the single high-order computer programming language that we have standardized for use in our defense systems.

I am here to reaffirm my commitment to the use of Ada throughout the Department of Defense. Ada is the language of choice for weapon system applications within the defense establishment. There are, however, many other areas where Ada can and will offer improved performance, enhanced coordination and communication, and greatly improve our capabilities.

We are very serious about using Ada-- considering what is at stake, I do not see how we could be otherwise. The services have already implemented policies and regulations for Ada use, and DoD will soon issue a directive requiring its use throughout the department.

My commitment to Ada is well-known, and the services are well along in implementing their use of Ada. The software industry is also aware of our commitment, and I note with pleasure that the number of validated Ada compilers has risen from just a handful two years ago to sixty-four as of today.

The key to developing high quality, reliable software for use by our defense establishment rests with each and every one of you. As you continue to work with Ada, I fully expect the industry will grow significantly. Indeed, the employment outlook for Ada software engineers is brighter now than at any time in the past.

The Ada software initiative is an example of America using its technological superiority. It was through the Ada program that we sought to create a high-order computer programming language that would satisfy the technical requirements imposed by all of our defense systems and which would ultimately replace the plethora of languages which had previously been in use within the Defense Department.

Without question, we have made significant progress in this program. The Ada programming language is not only a military standard, it has also been accepted as a Federal Information Processing Standard and by the American National Standards Institute. It will soon be accepted as an international standard.

While Ada is especially well-suited to the real-time requirements of our defense systems, its ability to meet the myriad requirements of almost all computer applications makes it attractive for a wide variety of other applications.

The success of the development of the Ada language was crucial to our software development initiative. Using Ada as its cornerstone, we established the Software Technology for Adaptable, Reliable Systems or STARS program, and we established the Software Engineering Institute, located in Pittsburgh. The goal of STARS is to reduce the time and costs normally associated with the development of defense software.

The primary goal of the Institute is to accelerate the transition of emerging software engineering techniques and methods into practice. This center for software excellence provides, within United States, a focal point for innovative research and development within the DoD software community.

The software initiative, through Ada, STARS, and the Software Engineering Institute, has been instrumental in energizing defense software technology. This software technology has, and will continue to have direct application, not only in U.S. defense systems, but in systems which we develop in cooperation with our allies. The Strategic Defense Initiative will build upon the technology, as will our cooperative armaments efforts with our allies.

What remains is a similar commitment from industry to provide us with the highest-quality Ada products now, for the success of Ada in the Department of Defense is yours to make.

Some years ago, the late General Secretary Brezhnev, in speaking to a Soviet people's congress, said: "In the competition between the two world opposed systems, the critical factor will be science and technology, and this makes major advances in science and technology of decisive importance." I might suggest that Mr. Brezhnev was only half right. He had the right strategy but he had the wrong country. The implementation of such a strategy demands technological innovation, which can only flourish in an environment of free expression and free enterprise. This is what Ada represents to the Defense Department and the nation.

Thank you very much.

APPENDIX C
LIST OF ATTENDEES

Brookshire, Jerry
Texas Instruments
PO Box 660246
MS 3114
Dallas, TX 75266

Clark, Peter
TASC
One Jacob Way
Reading, MA 01867

Crawford, Bard
TASC
One Jacob Way
Reading, MA 01867

Clow, Geoff
SofTech
16875 West Bernardo Dr.
San Diego, CA 92127

Facemire, Jeff
Texas Instruments
6550 Chase Oaks Dr.
PO Box 869305
MS 8435
Plano, TX 75086

Fanning, Shawn
SofTech
16875 West Bernardo Dr.
San Diego, CA 92127

Francl, Fred
Sonicraft, Inc.
8259 S. Greenwood
Chicago, IL 60618

Gicca, Greg
GTE Government Systems
1 Federal St.
Billerica, MA 01821

Harto, Debra
AFATL/DLCM
Eglin AFB, FL 32542-5000

Hazle, Marlene
MITRE Corp.
Burlington Rd.
Bedford, MA 01730

Jennings, Don
OL-ALC/MMECO
Tinker AFB, OK 73145-5990

Kean, Elizabeth
RADC/COEE
Griffiss AFB, NY 13441-5700

Lindquist, Tim
Arizona State University
Tempe, AZ 85287

Leavitt, Thomas
Boeing Military Airplane Co.
P.O. Box 7730
MS K75-67
Wichita, KS 67277

Maher, Patrick
Magnavox Electronic
Systems Co.
1313 Production Rd.
Ft. Wayne, IN 46808

Marmelstein, Lt Robert
AFWAL/AAAF-2
WPAFB, OH 45433-6543

Martin, Ronnie
Georgia Inst. of Technology
School of Information and
Computer Science
Atlanta, GA 30332

Mills, Mike
ASD-AFALC/AXTS
WPAFB, OH 45433

Mulholland, Sandi
General Dynamics
6100 Western Pl., Suite 735
Ft. Worth, TX 76101

Mumm, Hans
Naval Ocean Systems Center
Code 423
San Diego, CA 92152-5000

Myers, LCDR Philip
AJPO
Rm 3D139 (Fern St/C107)
The Pentagon
Washington, DC 20301-3081

Oberndorf, Patricia
Naval Ocean Systems Center
Code 423
San Diego, CA 95152-5000

Pritchett, Gary
SofTech, Inc.
16875 West Bernardo Dr.
San Diego, CA 92127

Reddan, John
SYSCON Corp.
3990 Sherman St.
San Diego, CA 92110

Rhoads, Barbara
Systran Corp.
4126 Linden Ave.
Dayton, OH 45432

Romanowsky, Helen
Rockwell International
400 Collins Rd. NE
Cedar Rapids, IA 52498

Shirley, Jane
Systran Corp.
4126 Linden Ave.
Dayton, OH 45432

Szymanski, Raymond
AFWAL/AAAF-2
WPAFB, OH 45433-6543

Stanton, John
AJPO
Rm 3D139 (Fern St/C107)
The Pentagon
Washington, DC 20301-3081

Weiderman, Nelson
Software Eng. Institute
Carnegie-Mellon Univ.
Pittsburgh, PA 15213

APPENDIX J

MINUTES

of the

EVALUATION AND VALIDATION (E&V) TEAM MEETING

4 - 6 MARCH 1987

The task for the Evaluation & Validation of Ada* Programming Support Environments (APSEs) is sponsored by the Ada Joint Program Office (AJPO).

*Ada is a Registered Trademark of the U.S. Government (Ada Joint Program Office).

TABLE OF CONTENTS

SECTION		PAGE
1.0	WEDNESDAY, 4 MARCH 1987	J-3
1.1	Welcome, Introductions and General Business	J-3
1.2	Software Development Environments Symposium Report	J-4
1.3	Ada Compiler Evaluation Capability (ACEC) Presentation	J-6
2.0	THURSDAY, 5 MARCH 1987	J-10
2.1	General Comments and Announcements	J-10
3.0	FRIDAY, 6 MARCH 1987	J-10
3.1	General Comments and Announcements	J-10
3.2	Working Group Reports	J-11
3.2.1	Coordination Working Group (COORDWG) Status Report	J-11
3.2.2	Requirements Working Group (REQWG) Status Report	J-11
3.2.3	Standards Evaluation and Validation Working Group (SEVWG) Status Report	J-12
3.3	AJPO Update	J-12
APPENDIX A	List of Attendees	J-14

1.0 WEDNESDAY, 4 MARCH 1987

1.1 Welcome, Introductions and General Business

The March 1987 meeting of the Evaluation and Validation of Ada Programming Support Environments (E&V) team was convened by Mr. Raymond Szymanski. He first introduced new team members Mike Burlakoff of Southwest Missouri State University, LT Mike LaPointe from Eglin AFB, and Christine Stacey from GTE. LT LaPointe will replace Debra Harto and Ms. Stacey will replace Greg Gicca, who has left GTE.

Mr. Szymanski informed the team of several Ada-related projects that have diverted his attention since the December meeting. The most significant one was an Ada workshop for the Avionics Laboratory in mid-January that required extensive pre-planning in December. On 23 February he briefed the Avionics Laboratory Board of Directors on the results of the workshop. The System Avionics Division has 25 Ada-related projects.

Twelve bidders submitted proposals for the Ada Compiler Evaluation Capability (ACEC) effort; the contract was awarded to Boeing Military Aircraft Company (BMAC) in Wichita. As a result, an ACEC working group will be formed during this session. This working group will be comprised of interested members from other working groups.

The Ada board's function was defined. It is comprised of Ada experts from industry who serve as technical advisors to the AJPO. Its members include Mr. Szymanski, Dr. Dudrey Smith from Lear Siegler, Dr. Ken Bowles of Telesoft, Dr. Jean Sammet with IBM, and Arnold Johnson from the National Bureau of Standards. They served as the lead panel on finalization of the Validation Procedures and Guidelines document. This group is known as the E&V Panel of the Ada Board, and volunteered their help to Mr. Szymanski, saying they would like to become more involved in E&V activities. Mr. Szymanski asked the Team to think of ways the Panel could help them and make note of their suggestions. This generated a comment by Sandy Mulholland. She stated her concern that the E&V Panel's involvement would foul communications between the Team and AJPO. John Stanton reassured the Team that no problems should arise from their involvement. Tim Lindquist stated that he viewed this development positively. Many past communications from the panel seemed to come from a misinformed viewpoint. More direct involvement from them should resolve that problem.

The Configuration Management plan has been approved and will be mailed out shortly. Mr. Szymanski noted that release letters for receipt of the Schema are necessary, and asked those people who hadn't yet signed them to do so. Mary Tompkins stated that these letters legally bind contractors without providing proper safeguards, and company lawyers would not advise signing them. Mr. Szymanski agreed that the letters could be drafted better, but they were an interim measure. He welcomed suggestions for improvement but stated that this measure was necessary for now.

Reorganization of the E&V Team along softer lines was discussed. By September, all members will belong to two working groups-- a directions WG and a product WG.

Mr. Szymanski then yielded the floor to Peter Clark and Bard Crawford.

1.2 Software Development Environments Symposium Report

Peter Clark, Bard Crawford
The Analytic Science Corporation

Mr. Clark opened the presentation by giving a status report on the Schema. He stated that the figures from the December meeting are now incorporated in Section 3.2, and that the Attribute Taxonomy was extensively rewritten based on the comments from REQWG. Then he turned the presentation over to Bard Crawford.

Dr. Crawford talked about a Software Development Environments Symposium that was held in January. The symposium is fully outlined in the ACM's Software Engineering Notes\& for January (ACM order # 548860, $14.85). He noted that the symposium had several areas that could be applied to the E&V's efforts. Since the ACEC and CVC will focus on two specific areas, he suggested that the team's attention to the whole-APSE should provide a balance.

In reviewing the symposium, Dr. Crawford noted that some well-integrated Software Development Environments (SDEs) were described in which extensibility is a major aspect. This is considered to be an attribute of APSEs. Lehman and Turski's paper on Essential Properties of IPSEs (Integrated Project Support Environments) defined them as "an embodiment of software technology in a collection of tools for capture, representation, control, refinement, transformation and other manipulation of project-related information." Dr. Crawford's assertion is that, despite the team's name (APSE E&V Team), "IPSE" expresses the subject of interest better than "APSE."

Necessary properties of IPSEs that could be termed "whole-APSE" attributes are structuredness, sufficient completeness, coherence/consistency, conservation of information, data structuredness, electronic office support, distributability and portability. An IPSE is evolutionary and extensible,adaptable to alternate methods of man-machine interface and advances in technology.

Houghton and Wallace's paper on Characteristics and Functions of Software Environments addressed environment types and the life cycle, integration, human factors, analysis and software quality, support for different user types, applications and hardware support, and differing levels of support. Under the topic "Environment Types", a general environment (such as ARGUS) is broken down into framing and programming environments. Each of these have their own distinct elements. The framing environment (e.g. SARA) is composed of initiation, definition, high-level and detailed design. Framing tools are typically more

specialized, and are methodology-specific. A programming environment (e.g. Stoneman/APSE) is made up of programming, operations and maintenance, and retirement.

"Levels of Integration" were represented as follows:

N Top (users)
[User Interface]

N-1 Intermediate level

N-2 Intermediate level

1 Intermediate level
[Machine Interface]

0 Bottom (machine)

The intermediate levels depicted are the software engineering environment. Using Stoneman as an example, there are three interim levels with the Common APSE Interface Set (CAIS) as the proposed standard interface to level 1.

A key issue of user interfaces with respect to integration is whether or not the environment keeps track of user activities and provides in-context services. This relates to "attribute names" in the taxonomy of the schema. "Analysis and Software Quality" has three elements that map to 2/3 of the Schema's functional taxonomy-- static analysis, dynamic analysis and management.

"Support for Different Types of Users" could be used as a model for the Schema's user index. Support is broken down as follows:

* "Organizers"
  Producer-- Manager
  Director-- Project Leader

* Others
  Designers
  Programmers
  Analysts
  Documentation Editor
  Librarian
  Maintenance Personnel

To quote from Houghton and Wallace, "An environment should orient its support to the player that is currently using [it]. . . other features should be hidden."

"Support for Applications" addressed Systems Development (hardware and software), Embedded Systems, Information Systems, Data Processing Applications and Security-critical Applications. This could be a viable

categorization for APSEs.

It was mentioned that, according to the National Security Department Directive (NSDD), all programming environments should have a C2 security level, as defined in the Orange Book, by 1990.

In conclusion, Dr. Crawford noted there is no universal set of attributes with comprehensive coverage of all aspects.

1.3 Ada Compiler Evaluation Capability (ACEC) Presentation

Thomas Leavitt
Boeing Military Aircraft Company

Mr. Leavitt opened his presentation by showing a roadmap of ACEC development. The BMAC test suite, AATPP, IDA (Institute for Defense Analysis) and BMAC analysis tools already exist and provide a basis for ACEC development. These will form a part of the test suite produced during Phase I of project development (ACEC version 1). In Phase I, critical performance criteria are identified and analyzed, prompting development of additional tests. Execution of tests along with a portability demo will produce data for a sample database in Version 1. Document tests will result in an interim technical report on this first version. Phase II comprises maintenance, enhancement and regression testing. Phase III takes the previous phase's results and develops additional tests. Phase IV follows with additional maintenance and more regression testing. The final result is ACEC Version 2, which will incorporate the test suite, analysis tools, sample database, guides and a report. Phase I is scheduled to last 18 months, and will end August 1988. Version 2 will be completed in March 1989 with the final version targeted for February 1990.

The presentation's scope was outlined as follows:

* Goals of ACEC
* Test Philosophy
* Tasks
* Test Suite
  - timing loop
  - statistical analysis
  - description of Test Suite
* Problems encountered so far
* Summary

The ACEC has four goals at its nucleus. First, it will serve as an aid to compiler selection by providing data that will allow the user to determine which is the best compiler for particular applications. Second, it is an implementor's aid that helps spot problem areas. The ACEC will prod implementors to support all tested features, and lastly, will aid application developers in estimating resources (coding styles and so on). The philosophy behind ACEC is to produce a product as portable and automated as possible, and whose results are repeatable.

The area on Tasks is the most detailed and encompasses four phases. Coordination under Phase I involved TIMS. review of existing test suites, documentation of existing tests and discussion with the Ada community. Identification and analysis of critical performance criteria encompassed studies performed by IDA, AATPP test suites, and a detailed study of operational Ada applications to isolate kernels.

Documentation of existing tests provides performance data as well as acceptable modifications that overcome new system limitations. Testing and evaluation have pointed out several problem areas. The first was in binary data handling, namely in error-correcting code and cryptographics. The need for a double-precision math library was identified, and the question of the package calendar's accuracy (compared to an external clock) was raised. Measuring Interrupt Response Time from an external signal to the ISR, as well as from an external signal to resumption of normal procession has become necessary. Compounded delay times caused by tasks running concurrently is a prime concern.

Also, tasking tests are an important part of Phase I. These tests measure time taken and report data types involved. Examples of these tests are for rendezvous, alternate selection, conditional select, types/numbers of passed parameters, exceptions in rendezvous, and when expression foldable clauses. Other tests are for conditional entry calls, task creation/termination, entry call tied to an interrupt, simple vs. complex accept, task attribute access and abortion.

Run-time tests and portability demonstrations are targeted for Ada compilers on DEC and Telesoft VAX, AIMS and Verdix 1750A, as well as the Harris and Gould self-targets. The tests perform statistical analysis on and comparison between these systems. Problems encountered in porting tests are being documented and may suggest modifications. An implementor's guide,reader's guide and technical report will be products of the report preparation process.

The task of developing analysis tools and documentation has resulted in three specialized programs-- MEDIAN, FORMAT and SPECIALIZE. MEDIAN allows comparisons between systems. FORMAT will extract timing data; however, Boeing's 1750 setup has no provisions for getting compiled output and this will cause some problems. SPECIALIZE is a tool that identifies changes that need to be made when porting the program to different targets.

The results of Phase I trigger Phase II, the maintenance and enhancement of ACEC Version 1 and regression testing. One problem uncovered already is that the tests aren't always testing what they're supposed to, and this is being corrected. Coordination in Phase III will consist of TIMS and reviews. Additional tests identified in Phase I will be implemented at this point and reports of progress and results will be made. In Phase IV, maintenance, enhancement and regression testing will be done on Version 2 of the ACEC.

Discussion of the test suite will cover four areas-- the timing loop, statistical analysis, current test set description, and portability. The loop will iterate a code fragment, making dynamic determinations of the number of iterations involved. The iterations employ the same computational path. The test suite keeps track of register allocations during this process and measures clock accuracy as well.

Statistical analysis identifies strong and weak points for each compiler and is statistically robust. The data model used in this process is:

Several different types of test problems are used-- classical benchmarks such as Wheatstone, Drystone, GAMM, and sort programs, among others; Ada versions of programs from the Computer Family Architecture study; specific optimization tests, programming kernels extracted from projects and FORTRAN inner loops. In addition, a Set of Simple Statement tests are used that study language features, identify for the same task affects performance.

Portability is an important issue. Portable test programs will run without modification or external equipment; however, not all language features can be tested in a portable way. For example, clock accuracy must be externally verified. Also, detailed measurements on some tests (i.e. interrupt processing) require additional equipment such as logic analyzers and signal generators. Overhead must be estimated with an external cyclic interrupt source. Also, testing system-dependent features necessarily limits portablility.

Several problems have already been encountered in various areas of Ada systems under test. Many systems are initially failing the tests. The best 1750A targeted system ran only 19 of 41 tests, while two other systems ran 3 of 41. Capacity limits were met with moderately-sized programs, and those programs using generics confused several of the systems. Some systems rejected arithmetic expressions as too complex. Problem areas likely to be uncovered in performance tests are dynamic storage reclamation and clock interrupts while raising exceptions.

After the presentation, the floor was opened to questions. The first question was about how the ACEC fits into overall Ada strategy, and if there would be an Ada Compiler Evaluation organization. John Stanton commented that test data should be collected and analyzed as compiler procurement will hinge on it. LCDR Myers then stated that the AJPO wants a perceived, credible process to evoke during evaluation. In the Navy, eight labs want these tests made. This is a policy, not task, issue.

There were several questions concerning the testing methods used. Someone noted the absence of PIWG tests and asked if they would be included. Mr. Leavitt answered no, and stated that the PIWG tests were continually evolving. If these were included, they would be quickly outdated. In the area of statistical analysis, a question concerning test prioritization was raised. Concern as to whether users can test only for their specific applications was expressed. Users will be able

to do this using run-time data included in the report. Another person pointed out that the tests aren't representative of real-world conditions. Some tests are unnecessary for certain users. Will only those tests required to check certain features be allowed? Mr. Leavitt answered that test modularity is under consideration. Another person asked if the ACEC would have to run all the tests, or if selected ones could be dropped. Mr. Leavitt replied that the user may pick and choose what is needed. Then he commented that some of the tests will be reviewed and dropped, but the remaining number will be between 800-900. Most are universal but some (like file I/O) aren't used in all versions and may be deleted.

Ronnie Martin asked if the ACEC would make allowances for limitations and performance problems in the processor chips themselves. She said that some chips may be able to handle certain functions better than others. Mr. Leavitt answered that everything figures in the evaluation--including the language definition, hardware, and operating system that Ada is running under. The chip factor is no more important than the other factors. This generated discussion and a comment that the end user would have to sit down, study the performance figures and determine whether Ada could be used with that compiler.

One person asked if different hierarchies of methods could be used in expression evaluation. Mr. Leavitt said that, in testing for optimizations, the concern is for valid programs. Then a question was asked concerning the nesting limit. Is the defined limit reasonable? Mr. Leavitt answered that reasonability limits will be decided by end users who complain to the vendors. The ACEC is not planned to be a stress test.

Time required to run the ACEC was asked about, and Tom stated that it takes around three hours (after compilation) to run on the VAX/VMS. He said this time is more or less the same on all systems except for 1750s.

A comment was made that some tests generate negative runtimes, and this generated a question as to whether across-the-board tests will take this into consideration and correct for it. Mr. Leavitt acknowledged a need for this but was unclear as to anticipated plans.

Someone asked if, considering the problems encountered on 1750A test results will be reported to the ACVC. Mr. Leavitt's response was that all test results were reported to vendors, some of whom made changes and are retesting their product.

One person raised a comment that he didn't understand who the tests are supposed to help. He defined himself as an end user who didn't care about anything but plugging in a software package and testing how efficiently it would run. These tests seem to be more for programmers than users. This generated discussion among the team members that this was a 'red-flag', and perhaps they were incorrectly defining that term. Tom stated that the ACEC generate subjective answers; there is no analysis involved, just result reporting.

LCDR Myers made a comment that the presentation's emphasis seemed to be on numerical results and were FORTRAN-oriented. He was concerned that the ACEC would not be testing Ada compilers that perform Ada, but Ada compilers that performed FORTRAN well. Mr. Leavitt said that the ACEC did indeed test Ada performance, although the presentation's emphasis seemed to be otherwise.

2.0 THURSDAY, 5 MARCH 1987

2.1 General Comments and Announcements

Mr. Szymanski announced that the APSE working group (chaired by Liz Kean) has disbanded. Their task is completed. He informed the Team that the ACEC working group would hold its first session at 2 p.m.

John Stanton noted that the ACEC wasn't only for the ABS-- third-party test houses would use it as well.

Following these brief announcements, the Team was dismissed into individual working groups.

3.0 FRIDAY, 6 MARCH 1987

3.1 General Comments and Announcements

Mr. Szymanski opened the session by reminding Team members to sign release forms for the Schema, Guide Book and Reader's Guides. He informed the Team that Ronnie Martin's project (STEP) has been completed, and this may be her last meeting. He thanked her for her participation and praised her contributions to the team's efforts.

The AJPO has asked the E&V team to justify their existence. Consequently, the team was asked to send comments on their perception of the usefulness of the E&V team to Mr. Szymanski within two weeks. Comments were to include what the companies involved (as well as team members) wanted to see as a result of the E&V team's efforts. Discussion ensued as to how these comments can help restructure and define the team's usefulness. Then Mr. Szymanski turned the floor over to general comments about LCDR Myers' retirement, resulting in a mini roast.

The need for additional time for Team meetings was discussed. Another contract would be under way by June, and the amount of time presently allotted was inadequate. Consequently Mr. Szymanski proposed a three full day format that would give the working groups another half day. Voting took place concerning which days the meetings should be held. Tuesday through Thursday received five votes, while the present schedule of Wednesday through Friday afternoon received eleven votes. Several members abstained as their future participation was in doubt and they didn't feel their votes should be counted. Then Mr. Szymanski proposed that only the June and September meetings be held in Dayton. Originally most of the E&V team members were from Dayton, but that is no longer

true. He felt that meetings in warmer climates would be more conducive to Team participation. General comments were favorable. LCDR Myers stated that meetings away from Dayton would incur extra costs for meeting rooms, and suggested that host companies might provide meeting facilities as General Dynamics had done in December.

Mr. Szymanski turned the floor over to Bob Marmelstein, who discussed Action Items from the first informal ACEC Working Group meeting. These are listed below:

ACEC Working Group Action Items	Responsible Person
1. Develop criteria for choosing beta test sites-- May 1987	R. Marmelstein
2. Guidelines for communication with BMAC-- May 1987	R. Marmelstein
3. Write summary of ACEC questions-- due April 1987	R. Marmelstein
4. Distribute Working Papers as required	T. Leavitt

LCDR Myers commented that Tricia Oberndorf, KIT chair, has had experience with forms and communications and suggested that LT Marmelstein contact her for information. Mr. Szymanski suggested that the team rewrite their ACEC questions and send them to LT Marmelstein over the NET.

3.2 Working Group Reports

3.2.1 Coordination Working Group (COORDWG) Status Report

Pat Maher delivered the COORDWG report. Their deliverable was the Public Coordination Strategy Document (PCS version 4) which is being completed. They reviewed the December minutes and discussed the issue of re-scoping their group. Action items for the next meeting include completing the PCS document, recruiting E&V team members, completing the status report, and revising the E&V Plan. Their deliverable due next quarter is the Technical Coordination Strategy document, version 4.0.

3.2.2 Requirements Working Group (REQWG) Status Report

Pat Lawlis reported that deliverables due include the Tools and Aids Document (version 1.0) and the Requirements Document (version 2.0). Accomplishments this meeting included commenting on the Tools and Aids and Requirements documents, and conducting a discussion of the E&V task's general statement of purpose. Key issues were the Requirements and Tools and Aids documents. Projected work includes revising and commenting on the Tools and Aids Document, revising the Requirements Document, and identifying information needed from government programs on their present and future needs for Tools and Aids. Deliverables due next

quarter are the updated version (2.0) of the Requirements Document, and version 1.0 of the Tools and Aids Document. No presentations are planned. The group made a formal request to Mr. Szymanski for future use of a PC by the group during their meetings.

3.2.3 Standards Evaluation and Validation Working Group (SEVWG) Status Report

The SEVWG report was given by Mike Mills. Their accomplishments included a new version of the CAIS Analysis Document, with the new title, "Issues and Strategies for CAIS Evaluation and Validation." This effort required minor changes, and will be considered an engineering release version. A version of this document addressing MIL-STD-1838A will be produced. Participation in a CVC working group was explored and possible future projects were outlined. The possibilities include addressing a graphics standard such as GKS, run-time system interfacing and involvement with DIANA. December SEVWG minutes were reviewed and approved. Comments from SofTech and the KIT were reviewed by the group for inclusion in the CAIS document.

At the conclusion of working group reports, Mr. Szymanski turned the floor over to LCDR Myers.

3.3 AJPO Update

LCDR Philip Myers
Ada Joint Program Office

LCDR Myers reported that the availability of DoD-STD-1838 has been delayed. The CAIS review board received more than 650 responses, and these must be answered. The document is camera-ready and should be out by the end of March.

The Department of Defense (DoD) is putting pressure behind the PDL, so should be a higher priority with the team. A DoD directive should be out soon, which will require the use of Ada in major new systems. An announcement about this directive's release will be made over the NET. The current DoD Directive on computer languages will be replaced. The use of C has never been a consideration because C is not standardized; however, Pascal is an approved language.

The NATO initiative is going well. Ten nations have signed technology exchange agreements, and two others are expected to do so. A Memorandum of Understanding (MOU) is in effect and provides for cooperative sharing of information. Tapes of two CAIS implementations (MVS and VMS) will be given to participating countries. Negotiations are in progress to establish the degree to which technology transfer will take place. Technical Advisory groups have been established. The International Standards Organization (ISO) standard of Ada is currently awaiting approval.

The AJPO has made Ada education a priority. An AFCEA study is due soon,

and establishment of Ada training requirements is being made. PC-based Ada compilers are being made available to universities on a free or affordable basis in a cooperative effort with the Software Engineering Institute.

AJPO. John Stanton will assume his duties on an interim basis. The new Ada Validation Guidelines (especially section 8) will be of particular interest to the E&V Team. These will address the documentation of Ada.

The last meeting of the retiring Ada board was held in February. The new board will consist of 20-25 members and is scheduled to be approved and installed by May. The major emphasis of this board will be runtime issues.

Much activity is taking place in the area of CAIS implementations. TRW began with a UNIX-based system but is currently porting it to VAX VMS. IBM is behind schedule in its development because a proper subset of 1838 has not yet been defined. The MITRE implementation has been completed. Their IR&D project ended last year, and development funds have been exhausted. However, MITRE's work demonstrated the viability of their approach. TCP is being partially used to distribute their version on SIMTEL-20. Gould is considering updating their version to comply with DoD-STD-1838. Performance will be the major emphasis of this development.

Adoption of Ada is well under way on several fronts. The first service to begin a strong Ada push is the Navy through their training initiative, but the Army isn't far behind in their efforts. The STARS CBD announcement is out; workshops are underway and three contracts have been issued.

LCDR Myers asked the team to share their perspective of Ada use. Mr. Szymanski stated that the System Avionics Division is beginning to implement more Ada. Bard Crawford (TASC) has observed the software industry's focus is shifting from labor to capital. Tom Leavitt (Boeing) has observed more production and less research. Mary Tompkins (Lockheed) and Pat Maher (Magnavox) noted there are more Ada-related projects to pursue. Mike Mills said more people are using Ada but leaving tasking to assemblers.

At the conclusion of the LCDR Myers' presentation, the March 1987 E&V team meeting was adjourned.

APPENDIX A

LIST OF ATTENDEES

Brookshire, Jerry
Texas Instruments
P.O. Box 660246
MS 3114
Dallas, TX 75266

Clark, Peter
TASC
55 Walkers Brook Drive
Reading, MA 01867

Combs, Phillip
Systran Corp.
4126 Linden Ave.
Dayton, OH 45432

Crawford, Bard
TASC
55 Walkers Brook Drive
Reading, MA 01867

Facemire, Jeff
Texas Instruments
6550 Chase Oaks Drive
PO Box 869305, MS 8435
Plano, TX 75086

Kean, Elizabeth
RADC/COEE
Griffiss AFB, NY 13441-5700

LaPointe, LT Mike
AD/ENE
Eglin AFB, FL 32542

Lawlis, MAJ Patricia
AFIT/ASU
3318 E. Dry Creek Rd.
Phoenix, AZ 85044

Lindquist, Tim
Arizona State University
Tempe, AZ 85287

Leavitt, Thomas
Boeing Military Airplane Co.
PO Box 7730, MS K765-67
Wichita, KS 67277-7730

Maher, Patrick
Magnavox Electronic
Systems Company
1313 Production Rd.
Ft. Wayne, IN 46808

Marmelstein, LT Robert
AFWAL/AAAF-2
WPAFB, OH 45433-6543

Martin, Ronnie
Georgia Inst. of Technology
School of Information and
Computer Science
Atlanta, GA 30332

Mills, Mike
ASD-AFALC/AXTS
WPAFB, OH 45433-6543

Mulholland, Sandi
General Electric
Mail Drop 913, GE/AESD
901 Broad S.
Utica, NY 13503

Myers, LCDR Philip
AJPO
Rm 3D139 (Fern St/C107)
The Pentagon
Washington, DC 20301-3081

Romanowsky, Helen
Rockwell International
400 Collins Rd., NE
Cedar Rapids, IA 52498

Shirley, Jane
Systran Corp.
4126 Linden Ave.
Dayton, OH 45432

Stacey, Christine
GTE Government Systems
1 Federal St.
Billerica, MA 01821

Szymanski, Raymond
AFWAL/AAAF-2
WPAFB, OH 45433-6543

Stanton, John
AJPO
Rm 3D139 (Fern St/C107)
The Pentagon
Washington, DC 20301-3081

Tompkins, Mary
Lockheed Austin Div.
Bldg. 30-E Dept. T2-32
2100 E. St. Elmo Rd.
Austin, TX 78744

Weiderman, Nelson
Software Eng. Institute
Carnegie-Mellon Univ.
Pittsburgh, PA 15213

Wills, Betty
CCSO/XPTB
Tinker AFB, OK 73145

APPENDIX K

MINUTES

of the

EVALUATION AND VALIDATION (E&V) TEAM MEETING

3-5 June 1987

The task for the Evaluation & Validation of Ada* Programming Support Environments (APSEs) is sponsored by the Ada Joint Program Office (AJPO).

*Ada is a registered trademark of the U.S. Government (Ada Joint Program Office)

TABLE OF CONTENTS

1.0 WEDNESDAY, 3 JUNE 1987 . . . . K-3
- 1.1 Welcome, Introductions and General Business K-3
- 1.2 ACEC Automatic Sizing Measurements K-4
- 1.3 E&V Technical Support Activities K-8
  - 1.3.1 The ARTEWG and E&V K-8
  - 1.3.2 The Reference System K-10
- 1.4 CAIS Validation Capability K-12

2.0 FRIDAY, 5 JUNE 1987 . . . . K-15
- 2.1 General Comments and Announcements, Morning Session K-15
- 2.2 AFSC Ada Task Team Report K-15
- 2.3 AJPO Report K-17
- 2.4 Working Group Reports, Afternoon Session K-18
  - 2.4.1 ACEC Working Group (ACECWG) K-18
  - 2.4.2 E&V Technology Classification Working Group (CLASSWG) K-18
  - 2.4.3 CAIS Validation Capability Working Group (CVCWG) K-19
  - 2.4.4 Standards Evaluation and Validation Working Group (SEVWG) K-19
  - 2.4.5 Requirements Working Group (REQWG) K-19
  - 2.4.6 Coordination Working Group (COORDWG) K-20

APPENDIX A Poem-- "Our Hero" (for Ray Szymanski) K-21

APPENDIX B LIST OF ATTENDEES K-22

1.0 WEDNESDAY, 3 JUNE 1987

1.1 Welcome, Introductions and General Business

The June 1987 meeting of the Evaluation and Validation (E&V) of Ada Programming Support Environments (APSEs) Team was convened by Mr. Raymond Szymanski. He opened by welcoming everyone and stated that the meeting marked his second year of involvement in E&V activities.

Mr. Szymanski next welcomed new Team members, Captain Bruce Hanna, Dan Eilers, Dave Davis, Linda Elderhorst, Tracy Holmes, and Sue LeGrand. Captain Hanna played a major role last year in revising the E&V Plan. Mr. Davis, an Air Force reserve officer from MITRE Corporation, is an AJPO representative to the NATO APSE Special working group. Sue LeGrand, of SofTech Houston, is program manager for the CAIS Validation Capability (CVC). Dan Eilers is the President and founder of Irvine Compiler, Inc., and Linda Elderhorst is from the Naval Air Test Center. Lastly, Tracy Holmes from GTE will replace Greg Gicca.

Guests included Captain Rebecca Abraham, Barbara Eldridge, and Frank Serna. Captain Abraham is with the Flight Dynamics Laboratory at Wright-Patterson, and is responsible for getting the AFWAL Commander to approve more Ada training for AFWAL. Ms. Eldridge, from the Avionics Laboratory at Wright-Patterson, was an observer. Mr. Serna , from TASC, gave a talk on ARTEWG activities as they pertain to the E&V effort.

The first topic of discussion was the distribution of Ada Compiler Evaluation Capability (ACEC) and Common APSE Interface Set Validation Capability (CVC) information. New cover sheets for these documents denote that distribution is limited to DoD components. This means there is no problem in distributing the information to government personnel, but contractors cannot access this information unless their companies are licensed or cleared to receive it.

Next, Mr. Szymanski talked about reorganization of the E&V Team. He noted that the Team is now operating in contractual and directional areas, and that the ACEC and CVC efforts will be taking a considerable amount of the Team's time. Mr. Szymanski asked John Stanton, the AJPO representative, how the AJPO viewed this shift in Team efforts. Mr. Stanton replied that the AJPO feels E&V efforts are making a significant contribution, and they have no problems with this division of attention.

Mr. Szymanski announced that review of the Reference Manual is being made by a handful of people outside the Team. These people are on the AJPO's E&V Panel of the Ada Board. Those reviewing the document include Dr. Dudrey Smith and Dr. Kenneth Bowles.

Mr. Szymanski then yielded the floor to Tom Leavitt of Boeing.

1.2 ACEC Automatic Sizing Measurements

Thomas Leavitt
Boeing Military Airplane Company

Mr. Leavitt opened his discussion by presenting reasons why automatic code sizing measurements are desirable. First, this method is faster in operation than other methods. There is less chance for human error in counting transcription errors and place-finding in listings. The most important reason for using this method is that it doesn't require the user to be intimately familiar with the ACEC.

There are several ways to achieve these goals. The first is to test program attribute size. However, this method is invalid as the language definition doesn't apply to programs, subprograms, blocks, and so on. The second method deals with address attributes. This involves bracketing the code you wish to size between two labels. This method assumes a linear code sequence and presents several problems including noncontiguous allocations of memory due to function declarations and declare blocks. For example, this method was used in testing six different compilers, only two of which supported the ADDRESS clause. One of the modes accepted the syntax but always returned zero as a result. Obviously this method was not useful for determining sizes, and this prompted development of an easier and more portable method.

The current ACEC design assumes that compiler portability will be compromised to the extent that a user can write a system-dependent function. This function will return the address of a calling routine, and will place the address where a function return result would go. Then the storage address may be found and returned. At this point, the code may be bracketed by two calls, allowing a subtraction to get the size difference. However, there are two constraints to this method. First, procedure prologues/epilogues and the code associated with computing the function result will get missed, unless the user is familiar with the code and places brackets in the right places. Second, measurement of the procedure call will be made if entry points are placed in the timing loop. This can be avoided by mechanically inserting the calls into the code, although this affects size.

After determining the sizing method, another area of concern relating to code size was discovered. Any piece of code has two distinct parts, the fixed and variable portions. The fixed part is comprised of the system's run-time and support libraries, while the variable part is made of user-written code. Estimating the fixed part is difficult as each system calls different external modules for the same function. Some compilers may, for the same feature, bring in modules that other compilers never call. This makes sizing determinations difficult. One way to estimate the fixed portion is to combine fixed portion be measured in a portable way? There is no answer to this question. The best non-portable method involves construction of a model program that reads the link map

and computes sizing information from it. Boeing's solution will be to write the model programs for certain systems and include these in the Implementor's Guide for that system.

Another complication of sizing involves systems that have shareable images, or multiprogramming systems. A large part of the run-time library is shared between all the systems, making the library's size variable. In addition, some things cannot be portably measured, such as the size and overhead of an activation record. However, this information can often be gleaned from reading the documentation-- writing a dedicated program to do it is unnecessary.

At this point the presentation disintegrated into a question and answer session and was never completed. Frank Serna commented that the "model" approach was flawed in its dependence on modules, link maps, and the like. Mr. Leavitt answered that this approach is very system dependent, and that is why it is such a problem. Then Mr. Serna interjected that some systems don't give detailed link maps-- what could be done then? Mr. Leavitt agreed and said that the first Ada system he examined would only return two answers, "correct link" or "failed during link." This required the user to write an analytic program to obtain more information. The only solution would be to obtain technical information that is only available to the implementor.

Tim Lindquist asked if it was possible to open an image file for analysis as a direct file of bits. Mr. Leavitt commented that most embedded systems don't allow I/O as they have limited memory space. Also, due to memory allocation, parts of programs can be located anywhere, while other areas may be reserved for stacks or heaps. Due to these considerations, the answer is no. Also, how can the user be sure he or she is actually looking at the image file?The file under examination may still include relocation information, flawing any analysis.

Mr. McKee asked how incremental costs in going from feature to feature are determined. Mr. Leavitt answered that worst-case program size is difficult to determine, even if all language features are utilized in the test program. This is due to implementation differences between compilers. Then Mr. McKee asked how the efficiency of Boolean representation was measured, and Mr. Leavitt stated that was determined by where the operands are declared. Mr. Lindquist commented that the first measurement option mentioned, size aptitude, doesn't apply to procedures and the like. Couldn't objects of task type be used, varying them in terms of what is placed in them? Mr. Leavitt said that was a possibility.

Mr. Serna commented that he felt dynamic allocation peaks were most important for embedded system users. Mr. Leavitt stated that Boeing's position was to not use dynamic allocation. This eliminates the chance of a program running out of storage space at execution time. Space requirements are computed beforehand and allocated statically. Someone commented that the definition of "static" could vary depending on the

particular optimization. Some optimizations may pull in only the external functions that are needed, making the static code size vary based on the routines called and the functions used. Mr. Leavitt answered that this point had been considered in the ACEC's design stage.

The next question concerned storage allocations-- what is being done to stress their dynamic behavior? There seem to be two issues. First, determining memory requirements in Ada real time systems is difficult. Second, predicting the behavior of real time systems themselves is difficult. Mr. Leavitt answered by relating a personal experience in this area. He said that tasking tests were in many systems, and this was due to excepts within SELECT statements. Vendors were allocating an object to keep track of open alternatives, and creating temporary storage during SELECTs. However, the storage wasn't deallocated afterwards, and looping in tasking tests caused their failure. This indicates a system bug not alluded to in the Language Reference Manual that should be fixed. Mr. McKee commented that he didn't see this as a major problem. A program might run many hours in a loop before failure, and this is not a common situation. He said that more thought should be given to this aspect.

Nelson Weiderman stated that he was unclear as to what was being tested for size. It seemed to him that one would need a "bag of tricks" available to test different features under different circumstances. LT Bob Marmelstein answered that the ACEC utilizes two basic types of tests-- one type tests for specific language features and the other tests overall application profiles. The language feature tests are used to measure the incremental cost of a given feature in an embedded target, among other things. At the very least, they want to determine the total size of the generated program.

Mr. Leavitt stated that he expects vendors to make information available to any customer about topics such as stack sizes. Mr. McKee commented that he would like to see a guide in future versions of the ACEC that reports which vendors willingly provide information and which ones do not. This prompted a discussion about attitudes assumed by ACEC users. Some disliked an adversarial posture, saying that compiler vendors should be trusted. Others said that blind trust was uncalled for, and that any product or performance specification should be viewed suspiciously.

Dr. Bard Crawford asked if the ACEC would merely contain test programs,or if comments would be made concerning documentation as well. Mr. Leavitt answered that no consideration was being given to this at the present time, and seemed uncertain as to future plans in this area. Then he commented that a number of things aren't addressed directly in the ACEC. For example, if a compiler aborts during compilation and the library is corrupted in the process, no ACEC feature evaluates this. However, a mechanism for noting problems and reporting them to the vendors exists in the ACEC.

Sandi Mulholland stated her concern that giving the ACEC to someone not knowledgeable about compilers will not assist them in evaluation. How is this being addressed? Mr. Leavitt said that two documents are included in the ACEC package-- an Implementor's Guide and a Reader's Guide. The first document is aimed toward people executing the ACEC, and is technically oriented. The second one is for end users, such as system buyers, programmers, project managers, and so on. A wide range of informational needs should be met with these two documents. Then Ms. Mulholland commented that it was important to have documentation on the theory behind evaluations to assist in understanding the answers. Mr. Leavitt answered that a tool is included to help evaluate differences between systems. However, this is an open-ended solution. Use of a code list and manual evaluation may be necessary to find the answers.

Mr. McKee voiced his disagreement with a stated aim of the ACEC. He that checking for specific optimizations is second-guessing compiler vendors and that this task should be delegated to IV&V, not the ACEC. LT Marmelstein answered his objection by stating that optimizations affect performance, so the ACEC is correct in testing for them.

Ms. Mulholland stated that unfair discrimination may result if the ACEC is slanted towards specific tests, such as for embedded applications. Some of the compilers with poor embedded target performance may do well for general applications. Mr. Leavitt said this was considered in ACEC design, prompting tests for direct and sequential I/O. These aren't always used on embedded systems.

Mr. Serna asked how the ACEC user would determine which tests were necessary to run for his or her application. Mr. Leavitt said that he expected people to run the whole ACEC suite. Depending on the system under test, some features may not work. The user would then write an error report to the implementor, then compose a report listing what did and did not work well. If a user didn't want to run the whole suite, sections may be chosen individually. An ACEC utility will indicate which tests are for which LRM sections.

Mr. McKee commented that he thought embedded systems were getting too much funding and attention. This area is a small part of the future compared with mini and mainframe computer usage. Ray Szymanski answered that money was being spent on embedded systems because they are the source of most problems, and commented that it was best to spend money on problem areas at this point. Mr. Leavitt commented there weren't many areas in non-embedded applications that weren't also relevant to embedded applications except for I/O-related areas specifically defined in the LRM. Instead, the focus should be on areas with unclear interpretations when comparing two compilers. Mr. Leavitt said that the ACEC isn't concerned with erroneous programs. Some people have the opinion that they should find all the LRM implementation options, test them and give the results to all users. This can encourage non-portable programming. The LRM states that expressions are not evaluated in a predefined order, but this may be changed at any time. However, the order in which tasks are accomplished can make performance differences.

This is why there are multiple test options available with ? common control path for comparison testing.

This prompted the question of how the term "erroneous program" is defined. Mr. Leavitt answered that any program that depends on an unspecified language feature is erroneous. For example, a program may depend on a compiler feature that has multiple solution paths. If the program works when one compiler takes a particular solution path but not when another compiler takes an alternate path, that program is erroneous.

This answer concluded the ACEC presentation.

1.3 E&V Technical Support Activities

Frank Serna, Dr. Bard Crawford, Peter Clark
The Analytic Sciences Corporation

1.3.1 The ARTEWG and E&V

Frank Serna briefed the E&V Team on TASC's activities with ARTEWG and how they relate to the Team. His presentation covered four areas, as follows:

Issues

- o Why the E&V Team should interface with the ARTEWG
- o Support of Cross-development E&V models
- o An example of Cross-development APSEs and Run-Time Environments
- o Recommendations to the Team.

Mr. Serna highlighted two of several important issues that should be examined from a whole-APSE perspective. First, the Run-Time Environment (RTE)ways-- tool structure and availability, nomenclature for describing the RTE, support of implementation dependencies and operating system interfaces. Second, cross-development is rapidly becoming the main APSE application. Many targets, such as the 1750A, may meet the system specification yet cannot support an APSE.

Next, Mr. Serna characterized a cross-development APSE. First, it must be integrated. Target selection must be simple, as it is with VAX XD-Ada. A software switch allows the programmer to change between native VAX mode and microprocessor mode. Second, transparent host-target migration is necessary to allow initial debugging on the host. This implies that the APSE handles and manages implementation dependencies. Third, a cross-development APSE should be able to list or evaluate implementation dependencies. This could be through tools that mark code or notify the user of the presence of implementation-dependent code. For example, the VAX Ada Portability Listing marks the code it processes in areas it perceives as non-portable.

Several recent ARTEWG products are particularly relevant to E&V. The first is a Canonical Model and Taxonomy of Ada Run-Time Environments. This defines the abstract machine interface and can be used as a source of RTE nomenclature also. The second product is a Catalog of Ada Run-Time Implementation Dependencies. This lists all features left up to the implementor and goes beyond the current scope of Chapter 13. This is necessary because dependencies impact portability not only through compilation but in execution behavior and performance. Also, dependencies impact APSE tools such as debuggers, linkers and loaders. A third document, Implementation of Run-Time Interfaces, contains suggested pragmas that access RTS features without redefining Ada or using non-portable machine-code insertions.

TASC is working with the Run-Time Canonical Model to develop an interface between the RTS and compiler-generated programs. The interface will assist with exception propagation and allocation of block-type storage. This is being accomplished through pragmas and packages, and is currently in draft form. The only well-developed part of the model addresses the interrupts.

Mr. Serna had several recommendations. The current E&V Reference Manual and Guidebook should be supplemented with a cross-development APSE model. Second, because the current Ada-Europe guidelines only touch on cross-development attributes for the compiler, E&V products should address the the remaining APSE tools. Third, a cross-development APSE model should be defined by a single paradigm.

This paradigm should:

- Use a canonical RTE model, such as the ARTEWGs (with modifications);

- Define host/target independent tools with the CAIS;

- Include simulator and emulator interfaces;

- Contain transport tools to assist code migration,

- Draw on APSE architectures from vendor implementations (such as DEC XD-Ada, Telesoft and Verdix).

The current Reference System contains isolated references to such topics as emulation, simulation/modeling, rehostability/retargetability and so on. The suggested Reference System should start with a separate cross-deveiopment formal index, or have a separate functional taxonomy group within the function index.

The ARTEWG has identified several RTE issues that are not currently addressed within their group, such as storage management, multi-programming, distributed processing, multi-level security, rollback-checkpoint recovery, performance monitors and garbage collection. These issues affect how APSEs operate and how they support targets in several

areas. These include multi-programming (in managing program library contention), the concept of a distributed APSE, and multi-level security within the APSE. These issues will have to be addressed by some group within the Ada community, as their impact is significant.

1.3.2 The Reference System

Dr. Crawford opened by briefly outlining the Document Review Schedule, then went on to discuss document relationships. The Reference Manual and Guidebook are collectively known as the Reference System. A document that helps make use of the Reference Manual is the E&V Classification Schema Report, distribution of which is limited to Team members and will not be released to the public.

The Reference Manual is divided into three main sections-- introductory chapters, reference material (chapters 4 through 7), and appendices. Each chapter of the reference material will correspond to an index in the Schema, and is broken down into a hierarchical taxonomy of elements. Each element has a number and name, and is composed of a text frame. This frame has a standardized format throughout the document, and includes a description, cross-references, and if applicable, guidebook references. The cross-references section includes areas for life-cycle phases and tools, and cross-references to other elements in other indexes.

Dr. Crawford explained how the documents relate to each other. Users will come into the Reference Manual from various places. They will then use the Reference Manual to obtain reference information or get pointers into the Guidebook for references to E&V technology. Internally there are five interrelated in different ways. The Function and Attribute indexes have important roles. First, the Function index is cross-referenced to all other indexes. Second, the Attribute index becomes important when E&V technology comes into play. After displaying a sample chapter layout for the Reference Manual, Dr. Crawford turned the presentation over to Peter Clark.

Mr Clark opened by saying TASC had received two responses after the March meeting concerning the Classification Schema. There were three areas addressed by these comments-- general style and punctuation, the revision schedule and the attribute taxonomy. Mr. Burlakoff was concerned about a statement in the schedule which said that the Schema would not be updated after AJPO approval. Mr. Clark explained that the Schema would not need to be as the Reference Manual and Guidebook would be updated as needed.

There were two opposing comments concerning the attribute taxonomy. Mr.Burlakoff stated that the top level was incorrect and that he liked the 2.0 version better. Ronnie Martin said the new taxonomy was a great improvement over version 2.0. Mr. Clark took the opportunity to illustrate the differences in the two versions. The old version had two categories, Functionally Dependent and Functionally Independent. The new taxonomy has three more descriptive categories-- Performance, Design and

Adaptation. This new version addresses shared criteria and is modular in design. The restructuring is based on an RADC document by Ronnie Martin, which says that many lower-level criteria support one or more quality factors at a higher level. For example, maintainability supports expandability, reusability, and so on. Mr. Burlakoff asked if mapping was used in comparing the two versions. Mr. Clark said yes, but some of the names were changed so this may not be evident. In summary, the old attribute taxononmy addressed functional dependency of attributes, ignored shared criteria, and was based on intuition. The new taxonomy ignores functional dependency of attributes, addresses shared criteria, and is based on previous work.

Mr. Clark then addressed changes from version 2.0 to 3.0 in the Reference Manual. Chapters 4 and 5, Life Cycle Phases and Tools, have no changes. Chapter 6, Attributes, has substantial changes. The only changes in Chapter 7, Functions, are the cross-references with the Attribute Taxonomy and inclusion of the ARTEWG's Canonical Run-Time Environment model. The Reference Manual's Attribute Taxonomy is now based on the RADC model for choosing applications software. Guidance in selection of attributes is based on four main criteria. The first area concerns system characteristics, or what attributes the user should be interested in. The second area deals with how low quality affects other factors. Shared criteria are addressed, and finally, beneficial and adverse relationships are explored.

Mr. Clark displayed a chart on factors related to system characteristics and briefly explained it. This chart was divided into two columns, showing relationships between application environment characteristics and software quality factors. Mr. Clark gave the following example using the chart as a reference. If the user were concerned about a long life cycle, he or she would look at the attributes of maintainability, expandability, and flexibility. Or, if the user were concerned about interactive systems, the attribute requiring attention is usability.

Next, Mr. Clark displayed an RADC chart on complementary factors. This chart showed that if a chosen tool is not correct, even though it provides consistent results its reliability is uncertain. Reliability, correctness, maintainability and verifiability are the most important attributes. The next area of concern is shared criteria. It is advantageous to assess shared criteria, as many benefits may be obtained from just one factor.

Finally, beneficial and adverse relationships were discussed. Although some criteria do not directly support certain factors, they may create an environment that is easier or more difficult to assess. For instance, usability is only directly supported by operability. But if a tool is very operable (i.e. user-friendly) maintainability is easier to assess. Also, tradeoffs may be indicated when assessing multiple attributes.

1.4 CAIS Validation Capability

Sue LeGrand
SofTech

Ms. LeGrand opened with some brief background explanation. CAIS is a standard interface between layers of an operating system. If the host is seen as the nucleus, it is surrounded by the Kernal Ada Program Support Environment (KAPSE). The CAIS surrounds the KAPSE, and is comprised of APSE- and user-supplied tools that facilitate interface with the KAPSE. It is the CAIS Validation Capability's (CVC) goal to build an extensive, re-usable test suite that validates implementations of the CAIS standard as it evolves.

Next, Ms. LeGrand went over the Phase I and II development schedule. Phase I began 4 May 1987 and extends until October 1988, and builds a test suite for DoD-STD-1838. Phase II incorporates enhancements and maintenance to Phase I and extends to October 1990.

Explanation of the CVC's technical approach was covered in two parts, the development approach and the framework and development sequence. The development approach incorporates four major steps-- analyzing requirements, collection of existing relevant tests, comparison of tests to produce a taxonomy, and test development and modification.

In the Analyze Requirements stage, inputs to the process are current CAIS documents, DoD-STD-1838, the Arizona State University (ASU) Operational Definition, the draft MIL-STD-1838A, and other sources offered by contacts. Outputs of this stage are a multi-level taxonomy of requirements and a list of required tests. The list is grouped into three levels of testing and three classes of test requirements. The methodology involved begins with an analysis of the CAIS documents. Requirements are extracted and translated into test requirements using a Common Test Requirements template. This template will be used throughout the CVC to achieve commonality. Then, test requirements are compared to 1838A documents for commonality and classified to see if the requirements are upgradeable as-is, upgradeable with modifications or not upgradeable at all.

Inputs to the Collect Existing Relevant Tests stage include ASU tests, MITRE tests, and others obtained from outside sources. Outputs include a catalog of tests that list test names and descriptions as well as level and class information, and computer files. The methodology used is to obtain tests and place them in a local database, then certify them against the description. The tests are then listed in a catalog according to the Common Test Requirements template.

Inputs to the process of Comparing Tests and Taxonomy will be the catalog of tests and the requirements in taxonomy format. The output will be a comparison matrix. This matrix will identify three levels and classes of tests, and a list of test requirements satisfied with existing tests. Obviously some areas of the CAIS will exist that

existing tests will not address. A list will reveal these holes to show where tests are needed, and this paves the way for the last stage of the development approach, Develop/Modify Tests. To accomplish this process, existing tests will be mapped into the Requirements taxonomy using the aforementioned template. This outputs a list detailing requirements fulfilled, tests to modify, and areas where new tests are needed. This allows test development to be prioritized. Also, some tests will be recommended to be deferred to the 1838A version, as current implementations may not include certain requirements of 1838A.

Ms. LeGrand terminated the presentation by briefly discussing the Framework and Development Sequence. It was described as being somewhat flexible, as new tests may be added at each level. These tests may be organized to spot non-conformities in CAIS implementations.

A question-and-answer session followed the presentation. Gary McKee wondered how much time was budgeted for CVC development. Ms. LeGrand said this was covered in the RFP milestones for the contract, and John Stanton commented that copies were available. Then Mr. McKee suggested that Ms. LeGrand could bring a list of CAIS documents that SofTech owns to the September meeting for the group's information.

Sandi Mulholland asked if the CVC would take European CAIS work into consideration. John Stanton answered that the CVC was not a required deliverable to the NATO effort, so any interaction between the U.S. and the Europeans would be minimized in this area. Then Mr. McKee referred to the Analyze Methods section of the CVC and asked if tests of detection mechanisms for exceptions were included. Ms. LeGrand answered yes, and when he suggested that boundary condition tests be included, she said they were.

The next question concerned procedures and operations-- what are the differences between them? Ms. LeGrand answered that operations consist of procedures-- procedures are the building blocks of operations. For example, if you wished to test an operation that opens, writes to, and closes a file, you must insure that the procedures to allow these functions are present. Then Tim Lindquist asked how the output from the first test stage will differ from the multi-level taxonomy that was outlined, and asked for an example of a test requirement. Ms. LeGrand answered the test requirements would fit into one of the output boxes as outlined. For an example, she mentioned the requirement to see if a procedure OPEN exists.

Next, Mr. Lindquist asked what was meant by capacity performance, and she answered that several tests weren't pass/fail, but a quantity measurement. This is what was meant by capacity. Marlene Hazle asked about plans to include a document that outlined test philosophy, concepts, and so on. John Stanton said this information was available in the proposal, or statement of work. However, this isn't commonly available. Ms. LeGrand stated that this information would be included in the Implementor's Guide and test Reader's Guide.

Mr. Lindquist commented that the ASU Operational Definition (OD) was being updated to 1838A, and validation was in progress. This prompted Fred Francl to ask how the CVC was different from the ASU validation process. Mr. Lindquist answered that their work was only validating the 1838A OD to insure it was worthy to be called an OD. Mr. Francl then asked if the ASU tests were as complete as SofTech's, and Mr. Lindquist answered that their work couldn't be considered a validation suite. Then someone asked for clarification: was the ASU validation only testing the OD, not CAIS? Mr. Lindquist verified this.

Mr. McKee asked if SofTech had or would get partial CAIS implementations to run CVC tests against. Ms. LeGrand said they would acquire any implementations that looked good enough to try. Mr. Stanton commented that he didn't feel comfortable with how tests would be identified and holes in testing methods located. Ms. LeGrand agreed and said that prudence was needed in going through requirements. Granularity would be carefully examined during this process.

Mr. Lindquist asked what impact single-user CAIS implementations would have on the CVC. Ms. LeGrand answered that Rich Thall, technical director of CAIS 1838A for SofTech, wants to accommodate every user from PCs to embedded systems in areas of security and distribution. The CVC's aim is the same. Marlene Hazle asked what the differences were between 1838 and 1838A, and how these differences would affect the CVC. John Stanton explained that 1838 was a "checkpoint" on the development path to 1838A. Mr. Szymanski commented that the U.S. is liable to the NATO effort for an international CAIS implementation, and the CVC may be used in testing this implementation before delivery, schedules permitting.

Mr. McKee commented on the differences between 1838 and 1838A, saying that development for each of these posed their own problems. He commented that SofTech should document their processes well enough that 1838A evaluation could be done rapidly and knowledgeably. This documentation should include such things as methodology, paradigms and taxonomies. This led Tim Lindquist to observe that there is a possible tenfold risk in going from 1838 to 1838A. One possible scenario is a five-year gap between these two standards that allows the creation of many good tool sets for the 1838. This makes CVC contract expenditures worthwhile, as there should be upward-compatibility between the two standards. The existing 1838 tools should work with modifications for the 1838A. TYPING on the 1838A can be set to allow it, and the I/O section will have compatibility packages that allow 1838 tools to run. These are good reasons to develop 1838 tools.

Ronnie Martin commented that the presentation was good, but she wanted assurance that suite evolution is carefully planned. She suggested that, if possible, the CVC be looked at from an incremental software development viewpoint with many tests along the way. Also, she said that many questions about test philosophy weren't answered. What types of tests will be used-- white box, black box, or some other type? Ms. LeGrand answered that this question would be answered later. Her presentation was preliminary and not designed for great detail.

Ronnie then asked what Ms. LeGrand's background was and what her previous involvement with CAIS had been. Ms. LeGrand said that she had belonged to the KIT/KITIA for two years as a member of the Requirements and Criteria working group (RACWG) for 1838A, and was a member of the Ada Board's Environment Panel. She was also a member of the 1838A review team. She had worked with Dr. Charles McKay at the University of Houston/Clear Lake and persuaded him to look at Ada for Space Station use. Previously she worked with Ford Aerospace and helped them design a new Local Area Network for the Johnson Space Center. During her work with McKay, she became interested in CAIS through his work with it as well as Ada and ISO/OSI.

At the termination of this discussion, tne Wednesday General Session was dismissed.

2.0 FRIDAY, 5 JUNE 1987

2.1 General Comments and Announcements, Morning Session

The general session opened with the reading of a poem about Ray Szymanski's softball skills as witnessed by several E&V'ers. Then the floor was turned over to Marlene Hazle for her report.

2.2 AFSC Ada Task Team Report

Marlene Hazle
MITRE Corporation

During March of this year, Ms. Hazle was a member of the AFSC Ada Task Team. This team was appointed to make recommendations to General Skantze on ways to facilitate Ada institution and usage within the Air Force. The team visited all of the AFSC product divisions and three laboratories, met with the AJPO and a STARS program representative, and was briefed by AFCEA personnel on their Ada training and education study. Team members were representatives from Air Force Systems Command and Logistics Command, AD, ASD, ESD/MITRE, and the NSIA. Ada use within various military programs is outlined below.

Ada is being used as ESD on the following programs:

- WWMCCS Information System (WIS)
- Joint Surveillance Target Attack Radar System (JSTARS)
- Sentinel Aspen
- Minimum Essential Emergency Communication Network (MEECN)
- Survivable Communication Interface System (SCIS)

- Command Center Processing and Display System Replacement (CCPDSR)

- Granite Sentry

Of these programs, the last four are NORAD/Cheyenne Mountain based. Sentinel Aspen, MEECN and WWMCCS all use Ada tasking, though Aspen uses tasking on top of other software. JSTARS uses Ada primarily as a Program Design Language.

In the Aeronautical Systems Division, 9 out of 51 programs currently use Ada. These include the Advanced Tactical Fighter, the AX 1750A compiler, SRAM II, and several other lab programs. AD uses Ada in the Modular Stand-off Weapons System. The SD had four programs including MILSTAR, and the BMO is using Ada for the Small ICBM program.

The team's goal was to talk to everyone actively involved in Ada, including program, engineering, and software managers as well as mission critical resource personnel for their impressions of Ada integration. They wanted to be aware of any impediments to Ada usage and to receive any suggestions for change. Their efforts uncovered six major problem areas. The first concerned policy-- requirements seemed unclear and/or conflicting. Next, there was concern over technical risk, specifically the combination of Ada and the 1750A. The next area of concern was over initial costs and schedule risks, seems to be a lack of planning data as well as a lack of process/procedures. This could be as simple as a set of documents outlining lessons learned, case histories, and so on. Finally, poor communication was named as a problem. Ada usage benefits are perceived to be in the future and for others, not those currently involved in the programs.

In the AFCEA briefing, the team was told that the capacity for Ada training and education was adequate and was available to Government contractors. There is a current surplus of Ada programmers as well. However, there is a significant lack of software engineering education across the board, and DoD training for management is inadequate. In summary, the programmers are trained but management is poorly trained in Ada utilization.

The team delivered their recommendations to General Skantze in a briefing on 15 April. First, they recommended the institution of an Ada Insertion Office at each product division within AFSC. This office would serve as an information clearinghouse and help center for those instituting Ada. It would assist SPOs in developing acquisition strategies, as well as policies and procedures to acquire Ada software effectively. They would also help SPOs with defining risk-management techniques and would participate in trade-off studies when a conflict was perceived. The Office would assist in waiver requests and consideration, and coordinate with their counterparts in other divisions. They would also be responsible for development and maintenance of a product division Ada Insertion Plan.

There were also specific recommendations concerning Logistics Command and AFSC. First, they should coordinate efforts at the product divisions and assure that clear and rational policies come from headquarters. The development and implementation of Ada risk-management activities for tool maturation is needed. Ms. Hazle added that the team's efforts were viewed by the team as important and were acted upon. Tool interface standards are needed, as well as the identification of flagship programs for special Ada support. Delegation of waiver control should be transferred from Air Force HQ to AFSC and Logistics Command. Air Force Ada training and educational needs should be vocalized, and an identification of courses necessary for Government management is needed. A request was also made for a reexamination of Ada usage on the 1750A. The 1750A will not meet all avionics needs and Ada must be compromised for use with it. General Skantze's reaction was positive; however, he requested data to prove Ada's benefits.

After this presentation, the Team was dismissed into individual working groups.

2.3 AJPO Report

John Stanton
Ada Joint Program Office

Mr. Stanton opened by announcing that the 3405.2 document had been signed. Document 3405.1 outlined all Ada language mandates, but was inadequate as work had stopped on SQL, the embedded query language for Ada databases. Since the document has been signed, SQL work will resume.

The AJPO is looking at all Ada related standards in greater detail, including the Graphics Kernel System (GKS), SQL, and perhaps PDL (program design language). They have approved funding for Technical Insertion Initiatives in the amount of $9 million dollars. This concept has been reviewed and approved by all three services and is expected to advance Ada integration, break down technical barriers, and aid in identifying new programs for Ada usage. One promising program already identified addresses the possible use of Ada in the area of Artificial Intelligence.

As of 1 June, 96 base Ada compilers have been validated, and this number was projected to reach 124 by 1 July. Version 1.9 of the ACVC was released on 1 June. The ACVC is currently on a six-month release schedule, but this may change to eighteen months after the inclusion of Chapter 13 tests.

After expressing the AJPO's satisfaction with the E&V's activities and level of participation, the discussion was turned over to working group reports.

2.4 Working Group Reports, Afternoon Session

2.4.1 ACEC Working Group (ACECWG)

Lt Bob Marmelstein gave the ACEC Working group report. After criticizing the group for a lack of requested feedback, he outlined the group's goals. First, they will provide a formal interface between the Ada community and the ACEC effort. Second, they will evaluate and critique aspects of the ACEC's technical approach and selected ACEC deliverables. Finally, they will discuss and provide feedback on ACEC-critical issues.

Next, Lt Marmelstein discussed present and future deliverables. Deliverables this quarter included a white paper briefing on executable code sizing measurements, evaluation of newly-released ACEC timing routines, and a draft of an ACEC usage summary that will be placed on the NET. Third quarter 1987 activities will include the release of documentation for 800 Boeing Military Airplane Company (BMAC) tests, production of a timing routines evaluation, and a review of ACEC requirements after the Requirements Specification is approved. Activities for fourth quarter 1987 include a white paper on ACEC test coverage and review of initial ACEC beta test site reports. Planned first quarter 1988 activities are an evaluation of ACEC Statistical Analysis methods and a review of ACEC beta test site interim reports. The second quarter of 1988 will provide a pre-release forum for the ACEC Version 1.

2.4.2 E&V Technology Classification Working Group (CLASSWG)

Ronnie Martin gave the working group's first report. There were no deliverables due as this was their first meeting. Accomplishments this meeting included development of a charter and analysis of Attribute Organization options.

The group's key issue was the handling of externally-developed technology. Projected work includes Reference Manual comments, review/analysis of select group comments, the Reference Manual's final review, investigation of options for the Guidebook's preliminary release, and the investigation of whole-APSE issues.

No deliverables are due next quarter, and no presentations are planned. However, Ms. Martin assigned two action items to the team. First, she solicited comments on the Reference Manual, with a deadline of 19 June. Second, she requested the team's input on the Guidebook.

Finally, Ms. Martin outlined the group's charter. They will serve as a focal point for analysis of the Reference System as embodied in the Reference Manual and Guidebook. They will solicit information and recommendations regarding E&V technology. The group will classify E&V technology and aid in technology transition in regard to the Reference System. Whole-APSE issues will be delineated, and new areas of investigation will be recommended.

2.4.3 CAIS Validation Capability Working Group (CVCWG)

Gary McKee gave this group's first report. Their accomplishments included the group's startup and organization, review of the CVC testing taxonomy, and establishment of their liaison position with the CVC contractor. Future plans include a September presentation by the CVC contractor on CVC testing methods and taxonomy, as well as the review of CVC contractor activities. Mr. McKee gave an Action Item to the team, and requested that everyone review MIL-STD-1838.

2.4.4 Standards Evaluation and Validation Working Group (SEVWG)

Gary McKee also gave this working group's report. This quarter's deliverable was the Issues and Strategies for Evaluation and Validation of CAIS Implementations document, version 1.0, given to Mr. Szymanski on 9 March. Accomplishments this quarter included establishment of the CVC working group and review of the CVC testing taxonomy. Future plans include a review of MIL-STD-1838 and the exploration of Ada display standards such as GKS, PHIGS windows. A presentation on MIL-STD-1838A by SofTech is planned for the future.

Action items include the gathering of information on Ada display standards, and review of the current 1838A draft. Mr. Szymanski was requested to provide information on portions of the 1838A presentation to be given at the upcoming KIT meeting to SEVWG.

2.4.5 Requirements Working Group (REQWG)

Marlene Hazle delivered the REQWG report. Their deliverable for this quarter was the Tools and Aids document, version 1.0. They completed the draft of this document, mapped the Requirements Document to the ACEC effort and explored future directions for the group. Projected work includes the delivery of the Tools and Aids document, version 1.0, and refinement of the group's future directions. This includes both technical areas and technology transition.

No deliverables are due next quarter and no presentations are planned.

Action items are as follows:

ACTION ITEM	RESPONSIBLE PERSON
Put status report on NET	H. Romanowski
Put latest Tools and Aids version on NET	J. Brookshire
Prepare white paper on ACEC/Requirements document mapping	R. Martin

Prepare short paper on ACEC issue of documentation of models/assumptions	D. Eilers
Put summary of UK MoD related activity on NET	N. Wiederman
Put future directions suggestions on NET	M. Hazle

2.4.6 Coordination Working Group (COORDWG)

Don Jennings delivered the COORDWG report. The main accomplishment this was a review of last quarter's minutes. Accomplishments this meeting were a review of the minutes, draft version 4.0 of the E&V Plan, production of the E&V Status Report, and exploration of reasons why the E&V team should continue. Deliverables due included the minutes, the status report, the Public Coordination Strategy (PCS) document, and the E&V Plan version 4.0. No presentations are planned for next quarter.

At the conclusion of Mr. Jennings presentation, the June 1987 E&V Team meeting was dismissed.

APPENDIX A

Poem

OUR HERO
(for Ray Szymanski)

In a game full of thrills galore,
 our hero reversed the score.
with the bases loaded
 his bat exploded--
you don't need to know any more.

Our hero struck a blow
 and the winning runs did flow.
He's a man of grace
 at second base--
but that's all you need to know.

He swung his mighty bat
 and that's all there was to that.
There's the story of
 his golden glove--
but never mind all that.

"The Bard" Crawford, et al.
E&V Team Meeting
June 5, 1987

APPENDIX B
LIST OF ATTENDEES

Belcher, Patty
SYSTRAN Corp.
4126 Linden Ave.
Dayton, OH 45432

Brookshire, Jerry
Texas Instruments
P.O. Box 660246, MS 3114
Dallas, TX 75266

Brown, Freda
SYSTRAN Corp.
4126 Linden Ave.
Dayton, OH 45432

Burlakoff, Mike
S.W. Missouri State Univ.
Computer Science Dept.
Springfield, MO 65804

Clark, Peter
TASC
55 Walkers Brook Dr.
Reading, MA 01867

Combs, Phillip
SYSTRAN Corp.
4126 Linden Ave.
Dayton, OH 45432

Crawford, Bard
TASC
55 Walkers Brook Dr.
Reading, MA 01867

Eilers, Dan
Irvine Compiler Corp.
10821 Skypark Cir., # L
Irvine, CA 92714

Elderhorst, Linda
Naval Air Test Center
Code SY31H
Patuxent River, MD 20670

Eldridge, Barbara
AFWAL/AAAS-2
WPAFB, OH 45433

Facemire, Jeff
Texas Instruments
6550 Chase Oaks Dr.
P.O. Box 869305, MS 8435
Plano, TX 75086

Fainter, Robert
Computer Science Dept.
Arizona State University
Tempe, AZ 85285

Francl, Fred
Sonicraft, Inc.
8859 S. Greenwood
Chicago, IL 60619

Froidl, Jack
AFWAL/AAAF-2
WPAFB, OH 45433

Hanna, CAPT Bruce
AFWAL/AAAF-2
WPAFB, OH 45433

Hazle, Marlene
MITRE Corp.
Burlington Rd.
Bedford, MA 01730

Holmes, Tracy
GTE Govt. Systems
1 Federal St.
Billerica, MA 01821

Jennings, Don
OC-ALC/MMECO
Tinker AFB, OK 73145-5990

Kean, Elizabeth
RADC/COEE
Griffiss AFB, NY 13441

LaPointe, Lt Mike
AD/ENE
Eglin AFB, FL 32542

Lawlis, MAJ Patricia
AFIT/ASU
3318 E. Dry Creek Rd.
Phoenix, AZ 85044

Leavitt, Thomas
Boeing Military Airplane Co.
P.O. Box 7730, MS K75-67
Wichita, KS 67277-7730

LeGrand, Sue
SofTech
1300 Hercules Dr., Suite 105
Houston, TX 77058

Lindquist, Tim
Computer Science Dept.
Arizona State University
Tempe, AZ 85287

Maher, Patrick
Magnavox Elect. Systems Co.
1313 Production Rd.
Ft. Wayne, IN 46808

Marmelstein, Lt Robert
AFWAL/AAAF-2
WPAFB, OH 45433

Martin, Ronnie
Georgia Inst. of Technology
Software Eng. Research Ctr.
Atlanta, GA 30332-0280

McBride, John
SofTech
1300 Hercules Dr., Su. 105
Houston, TX 77058

McKee, Gary
Martin Marietta
Information & Communication
MS L1640, P.O. Box 179
Denver, CO 80201

Mills, Mike
ASD-AFALC/AXTS
WPAFB, OH 45433

Mulholland, Sandi
General Electric/AES9
Mail Drop 860
901 Broad St.
Utica, NY 13503

Oberndorf, Patricia
NOSC
Code 423
San Diego, CA 95152-5000

Pickart, CAPT Bruce
HQ AFOTEC/LG5S
Kirkland AFB, NM 87117-7001

Reddan, John
SYSCON Corp.
3990 Sherman St.
San Diego, CA 92110

Romanowski, Helen
Rockwell Intl.
400 Collins Rd. NE
Cedar Rapids, IA 52498

Schoonover, MAJ Kenneth
HQ AFSC/PLRT
Andrews AFB, MD 20334-5000

Shirley, Jane
SYSTRAN Corp.
4126 Linden Ave.
Dayton, OH 45432

Stanton, John
AJPO Rm 3D139
(Fern St./C107)
The Pentagon
Washington, DC 20301-3081

Szymanski, Raymond
AFWAL/AAAF-2
WPAFB, OH 45433

Wiederman, Nelson
Software Eng. Institute
Carnegie-Mellon Univ.
Pittsburgh, PA 15213

Williamson, James
AFWAL/AAAF-2
WPAFB, OH 45433

Wills, Betty
CCSO/XPTB
Tinker AFB, OK 73145

END
DATE
FILMED
9-88
DTIC